Visualization with 3D Engine

KOREA UNIVERSITY

KUCG

Contents

- **Assignment #3**
  - 3D engine으로 Robot Arm 제어
    - **Shading Method(Normal Mapping, Environment Mapping)**
    - **Hierarchical control of Robot arm**
- **3D Engine: 다누리VR**
  - **Install & User Interface**
  - **Shading & Control**
    - **Project 생성**
    - **Scene 구성 : Object 생성 및 제어(카메라, Light, Object,,,, etc.)**
    - **Shading: Texture Mapping 등**
    - **Scene control with Python**
      - **Basic Transformation: Translation, Rotation, Scaling**
      - **Hierarchical Object Transformation**

Assignment #3

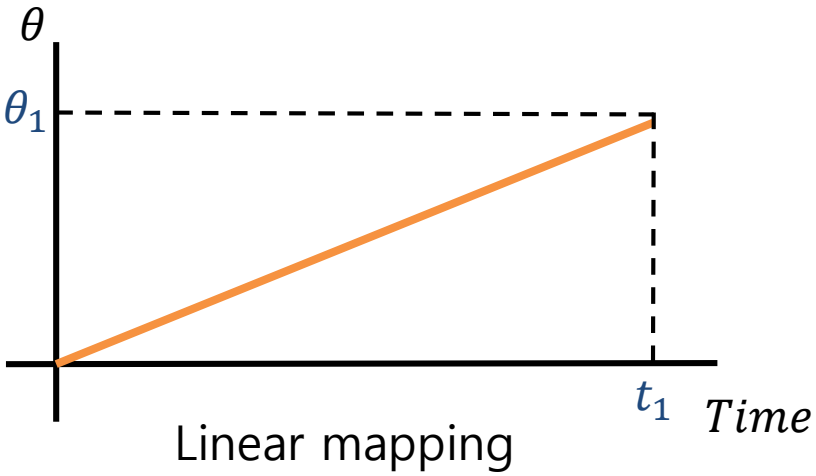
Assignment #3

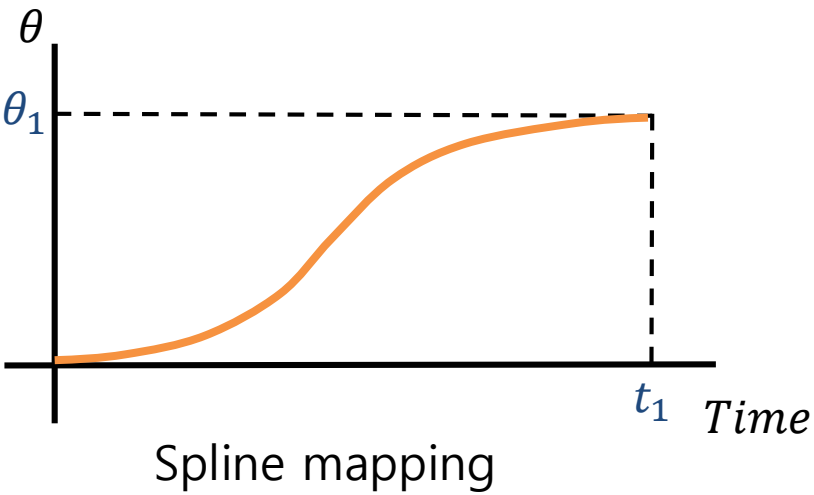
VR engine으로 Robot Arm 제어

- Requirements
  1. Shading Method
     - Normal mapping with your name and ID.
     - Environment mapping with sky Box.
  2. Hierarchical control of Robot arm
     - Implement a 3 degree of freedom robot arm which is composed of Upper Arm, Lower Arm and Hand.
     - Control the angle of joints by Keyboard callback.
     - Using Python code to control the motion.
- Additional score
  1. Realizing natural motion
     - Control the angle of joints by Update callback.
     - Applying appropriate time-angle mapping.

Additional score: Time-angle mapping.

Spline mapping looks more natural



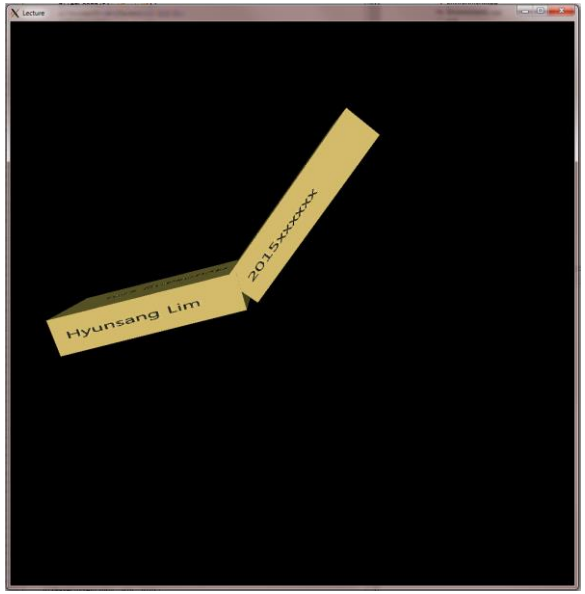


2차 숙제와의 다른점

(1) Using VR Engine(다누리 VR) instead of OpenGL

(2) Synchronous control of Three joint angles.

(3) Implementing Environment mapping



[Assignment2 Image]

Result Example

Result Example (Synchronous control of angles)

Submit the Assignment

- Submit the zip file @ Blackboard
  - File name must be “Assignment3_StudentID_Name.zip”
    - Ex. Assignment3_2015000000_박지혁.zip
  - Must include
    - Src file
      - Danuri projectfile(Including Python code)
    - Result running video file
  - Due date: November 19th

3D Engine: 다누리VR

다누리VR

Install & User Interface

Install Link

- http://211.230.48.42/g5/

Danuri VR download page

정보수정 로그아웃

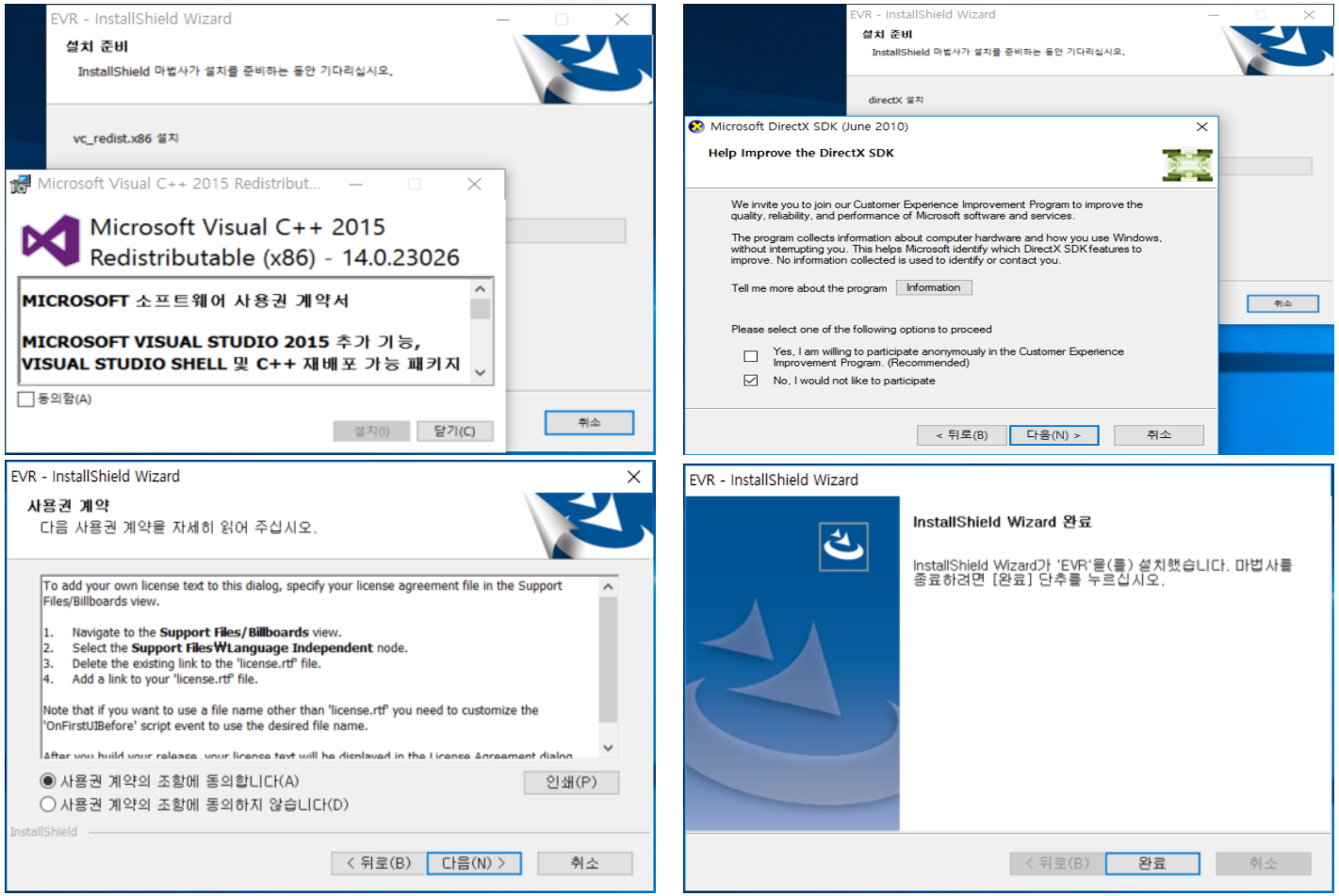
다누리 DANURI-VR

개요 | 다운로드 | FAQ | 라이선스 | 활동

소프트웨어

데이터 종류	버전	다운로드	다운로드 수
다누리-VR 실행파일	1.0 _ 2017-06-30	LINK	44
다누리-VR 소스파일	1.0 _ 2017-06-30	LINK	25

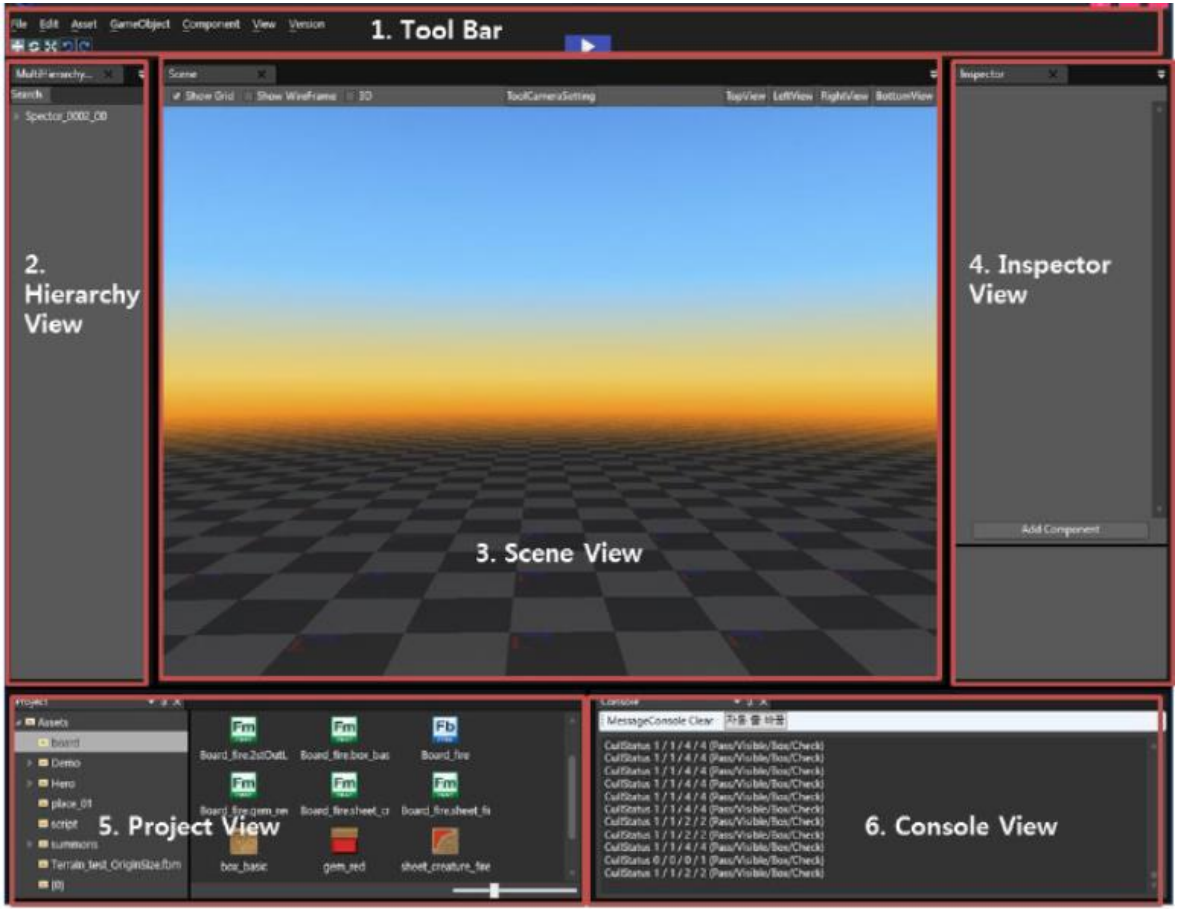
Install procedure



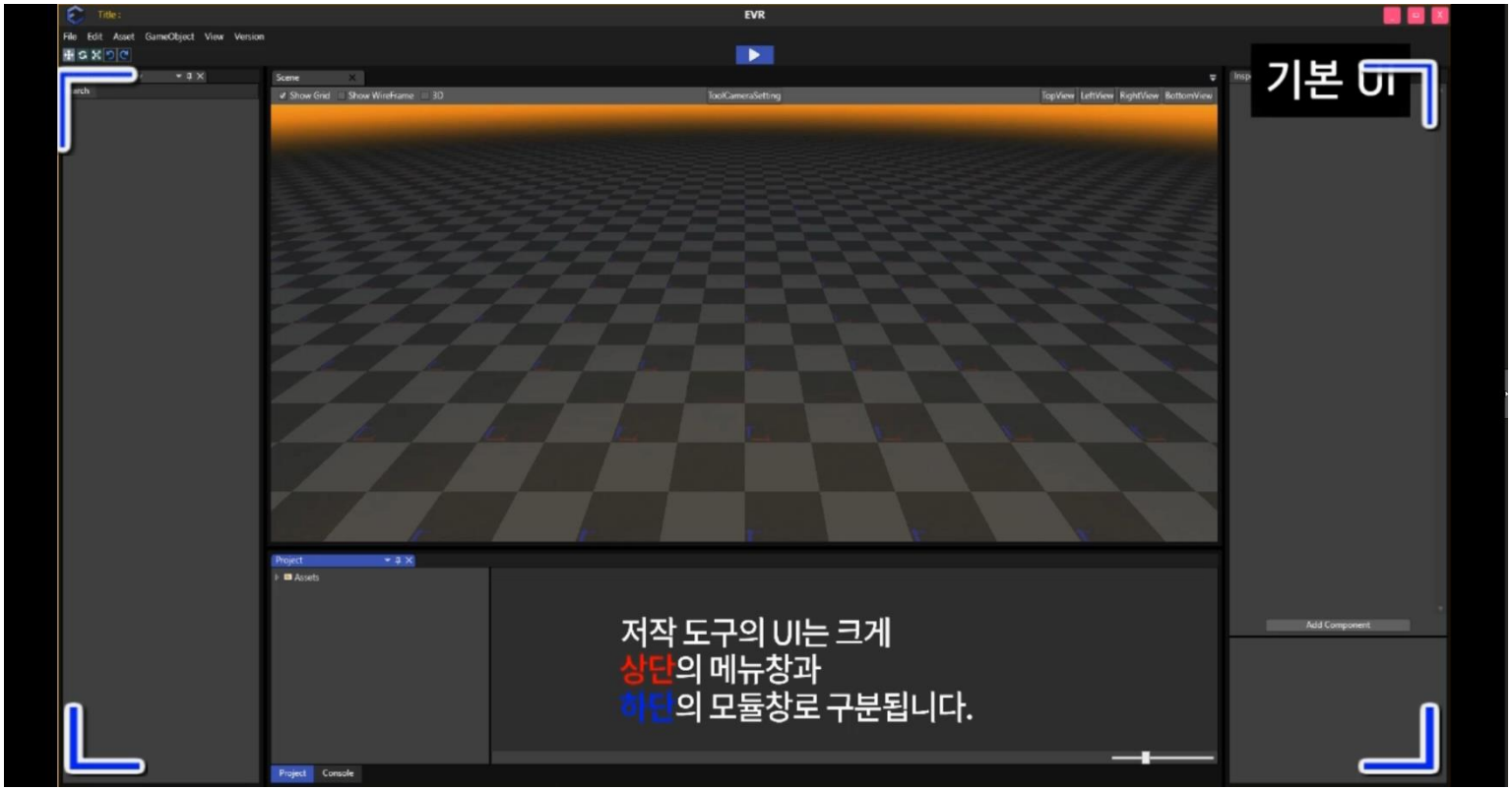
- 다누리-VR 저작도구 설치 완료

UI - Overview

- 화면구성
  - ToolBar
  - Hierarchy View
  - Scene View
  - Inspector View
  - Project View
  - Console View

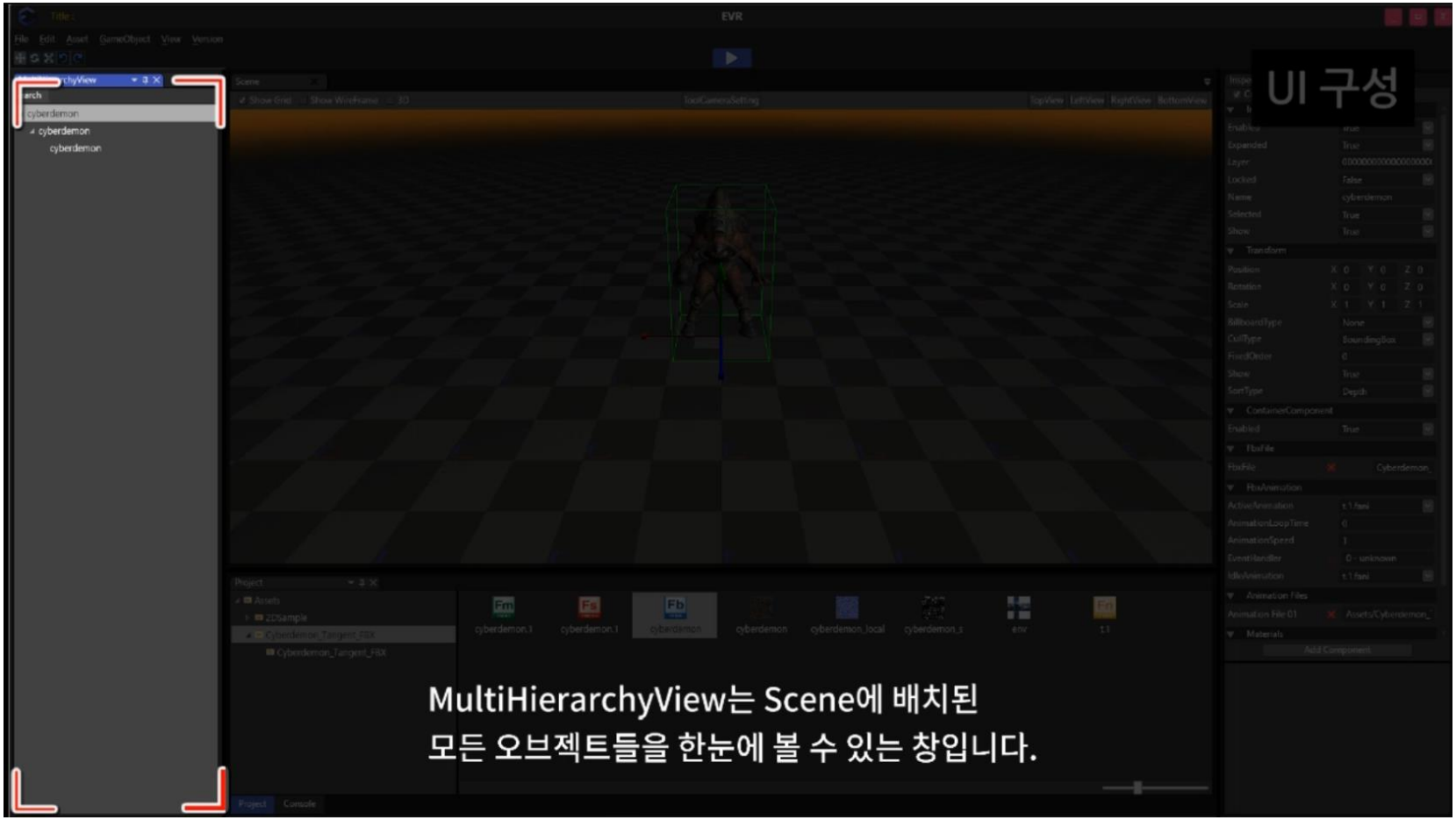


UI Composition



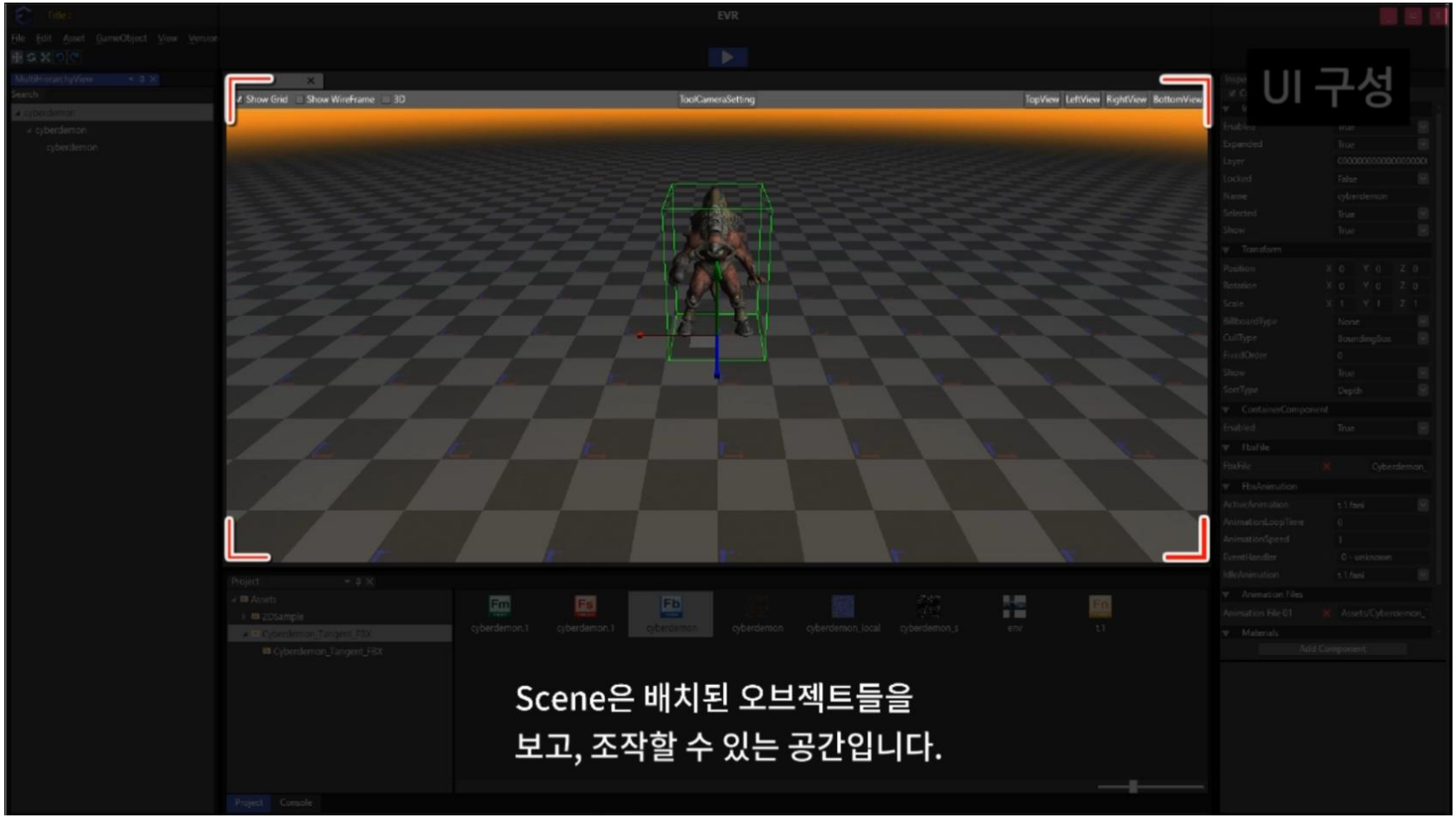
- UI 구성
  - 상단의 메뉴창과 하단의 모듈창으로 구분됨

Multi Hierarchy View



- 오브젝트들의 Hierarchy 구조 확인 및 변경 가능

Scene

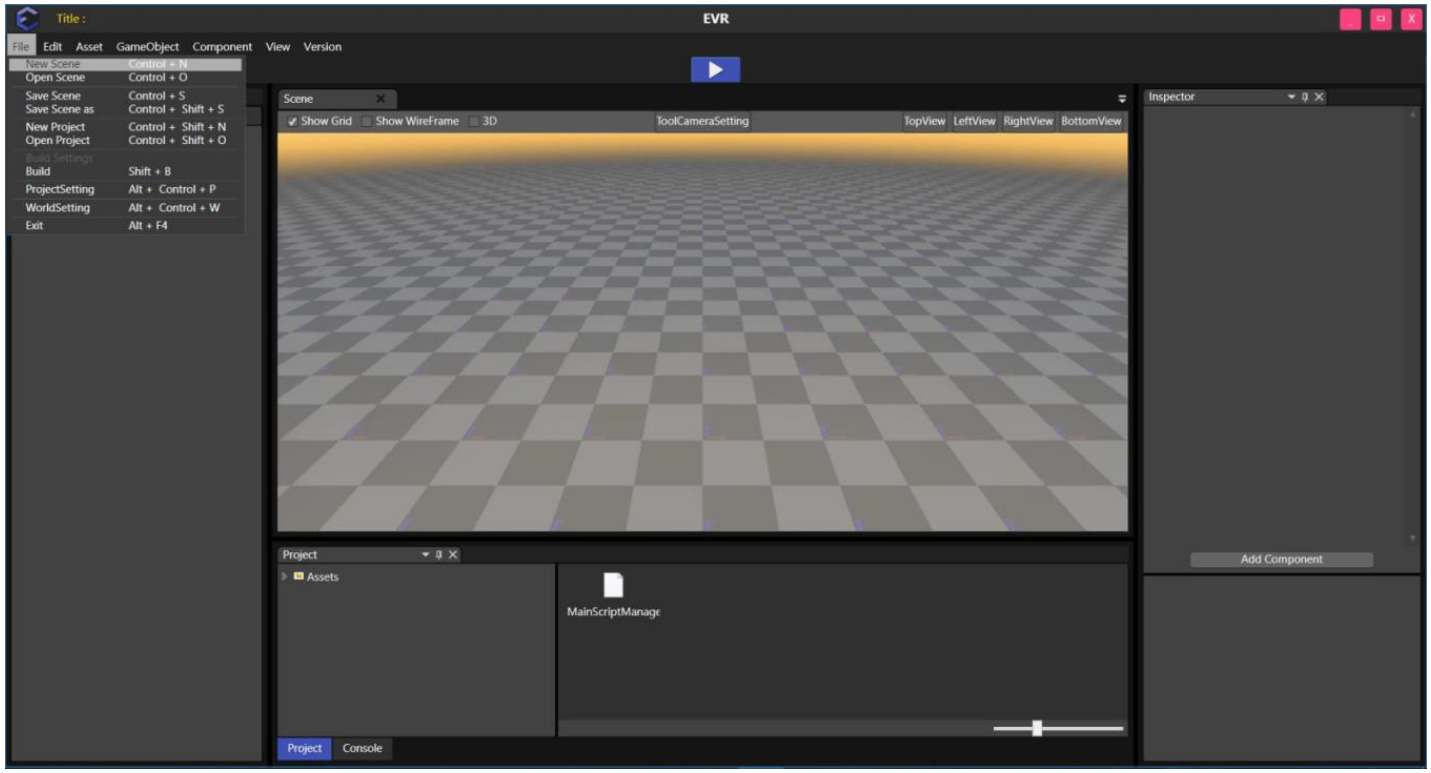


Inspector

Project

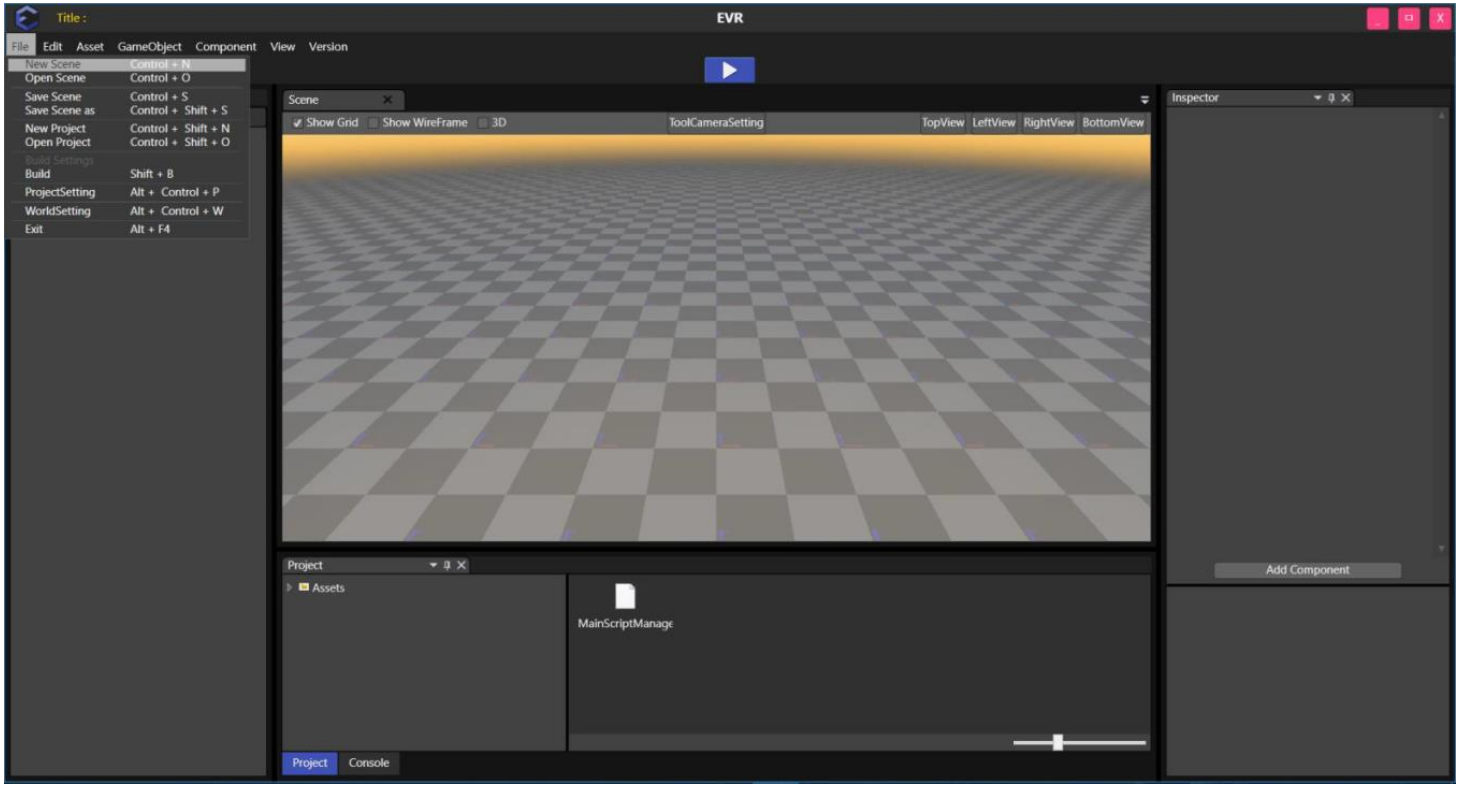
[Asset창 우클릭] – [OpenExplorer] 한 후 Asset file import 가능

Toolbar – File Ⅰ



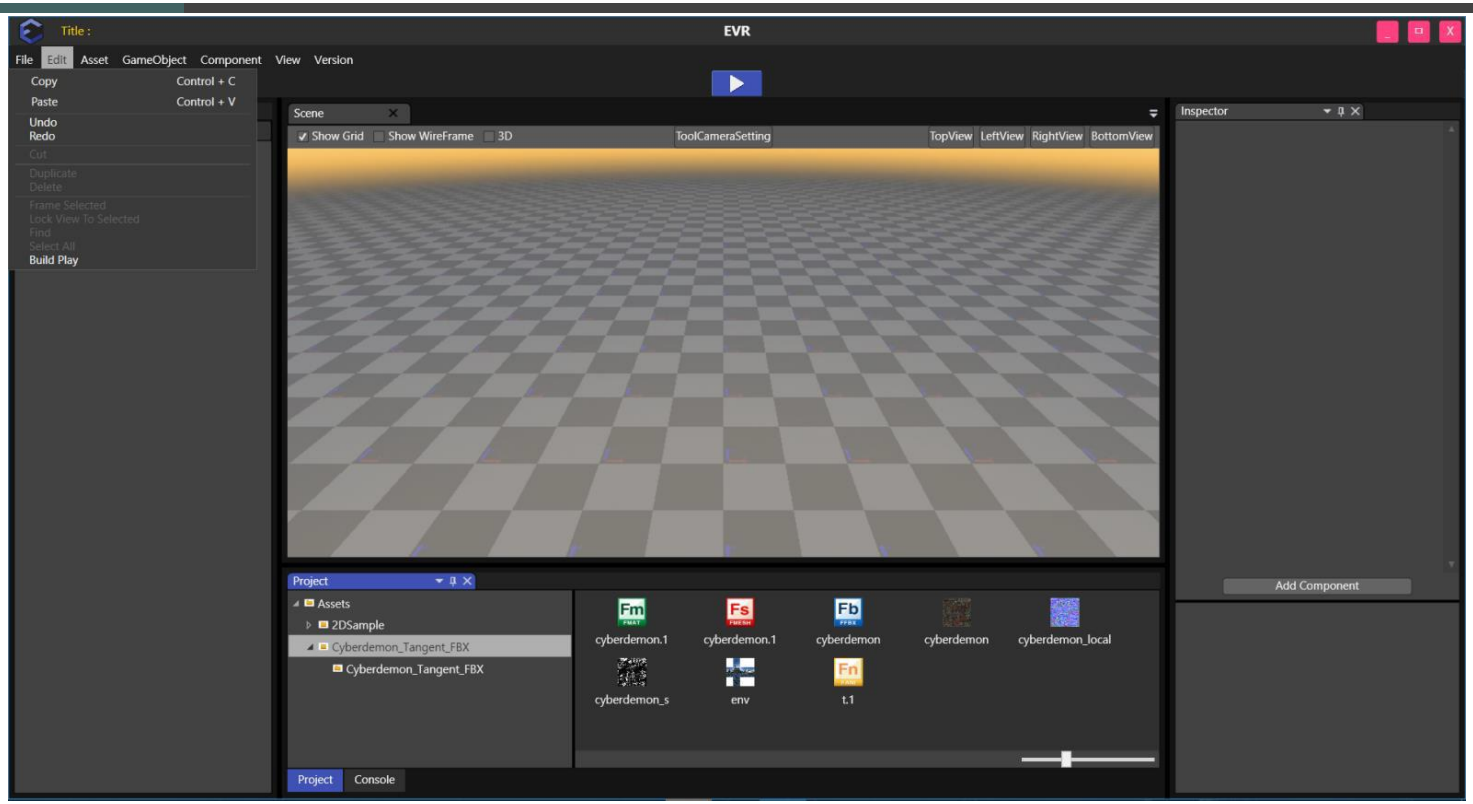
- New Scene: 새로운 씬을 생성
- Open Scene: 기존에 존재하는 씬을 불러옴
- Save Scene: 제작중인 씬을 저장
- Save as Scene 제작중인 씬을 다른 이름으로 저장
- New Project: 새로운 프로젝트를 생성

Toolbar – File Ⅱ



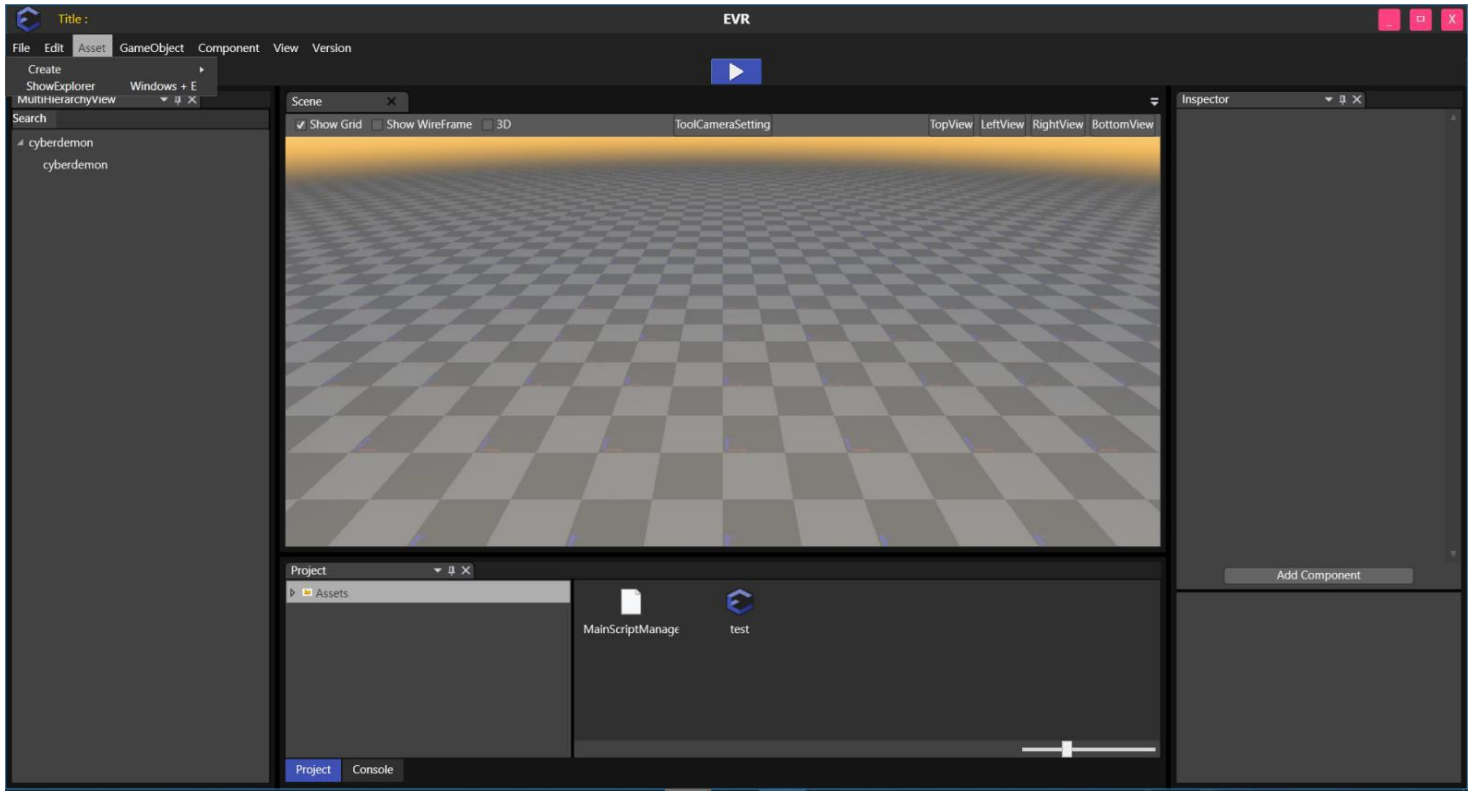
- Open Project: 기존에 존재하는 프로젝트를 불러옴
- Build: 작업중인 씬을 빌드
- Project Setting: 프로젝트의 설정을 조절
- World Setting: 월드 컨테이너를 설정
- Exit: 프로그램을 종료

Toolbar - Edit



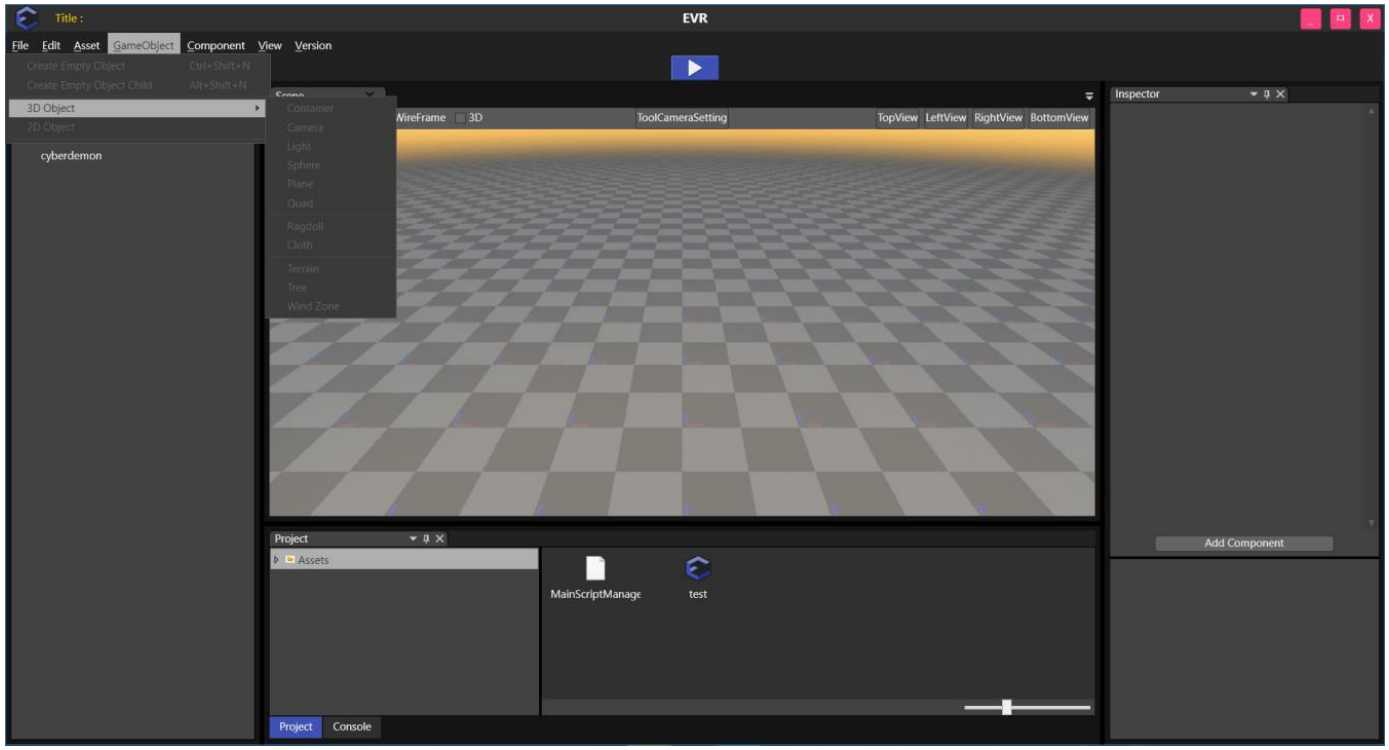
- Copy: 객체를 복사
- Paste: 복사한 객체를 붙여 넣음
- Undo: 최근 작업을 취소
- Redo: Undo 작업을 취소
- Build Play: 제작한 씬을 실행

Toolbar - Asset



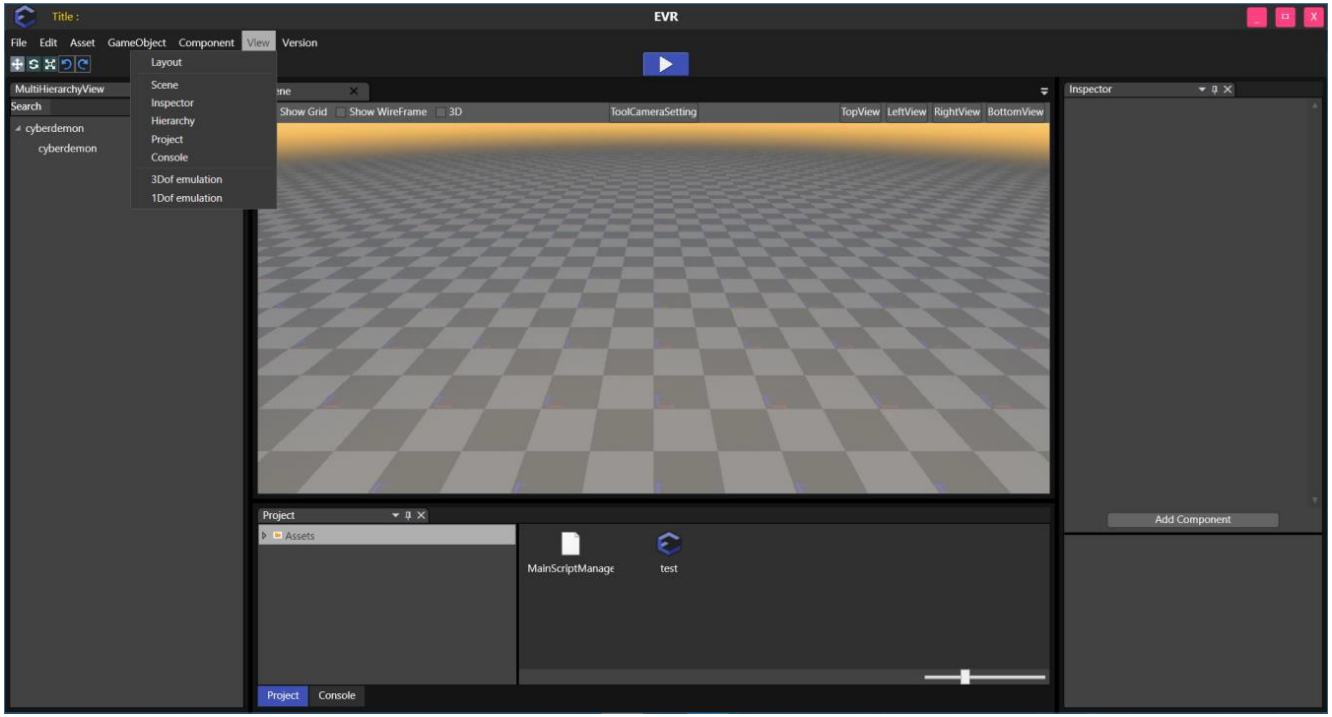
- Asset 메뉴
  - Create: 에셋 폴더 내에 새로운 에셋을 생성
  - ShowExplorer: 에셋 폴더를 파일 탐색기로 실행

Toolbar - GameObject



- GameObject 메뉴
  - Create Empty Object: 빈 오브젝트를 생성
  - Create Empty Object Child: 빈 자식 오브젝트를 생성
  - 3D Object: 3D 오브젝트를 생성
  - 2D Object: 2D 오브젝트를 생성

Toolbar - View

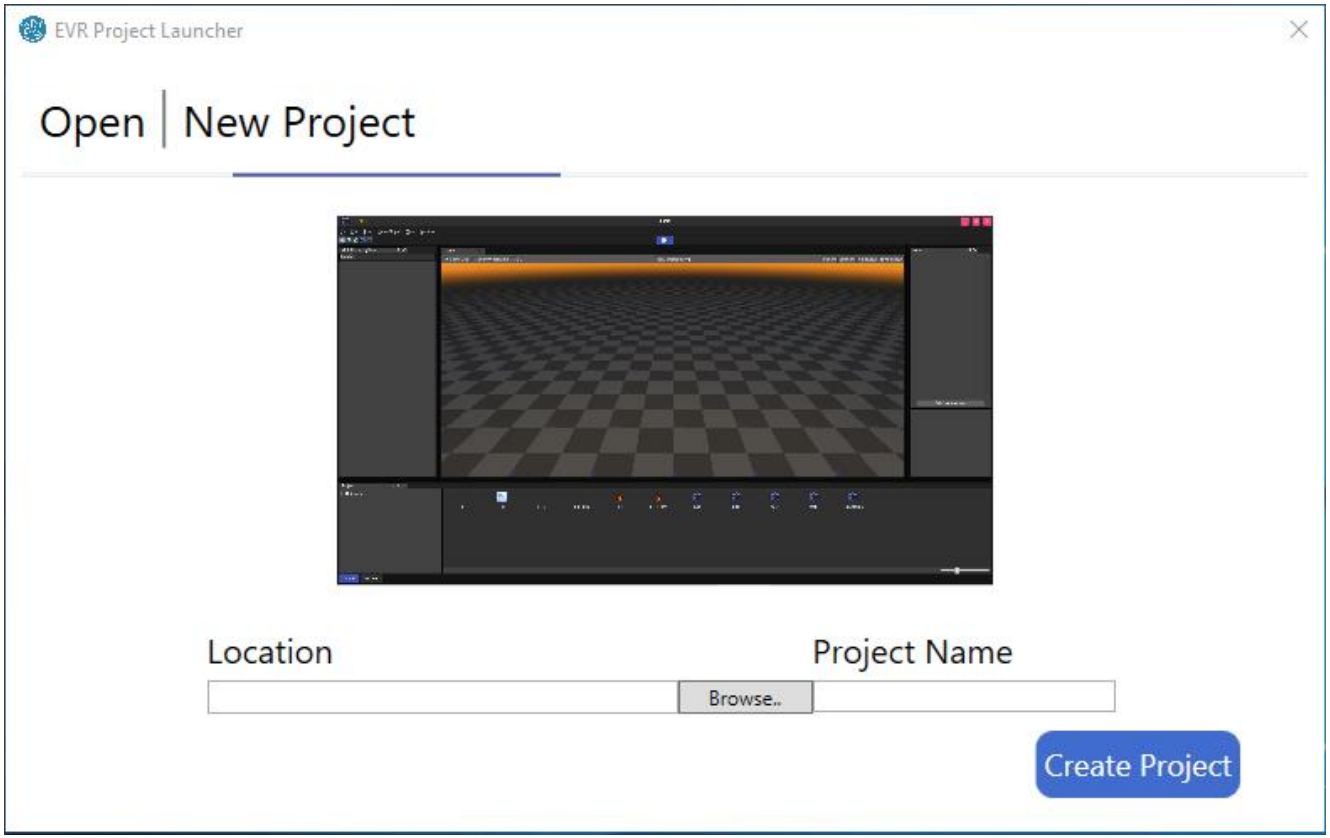


- View 메뉴
  - Layout: 화면 레이아웃을 설정
  - Scene: Scene View를 설정
  - Inspector: Inspector View를 생성
  - 3Dof emulation: 3Dof Emulator를 실행
  - 1Dof emulation: 1Dof Emulator를 실행

다누리 VR: Shading & Control

Project 생성

Create Project



- 프로젝트 생성
  - New Project -> Location 설정 -> CreateProject

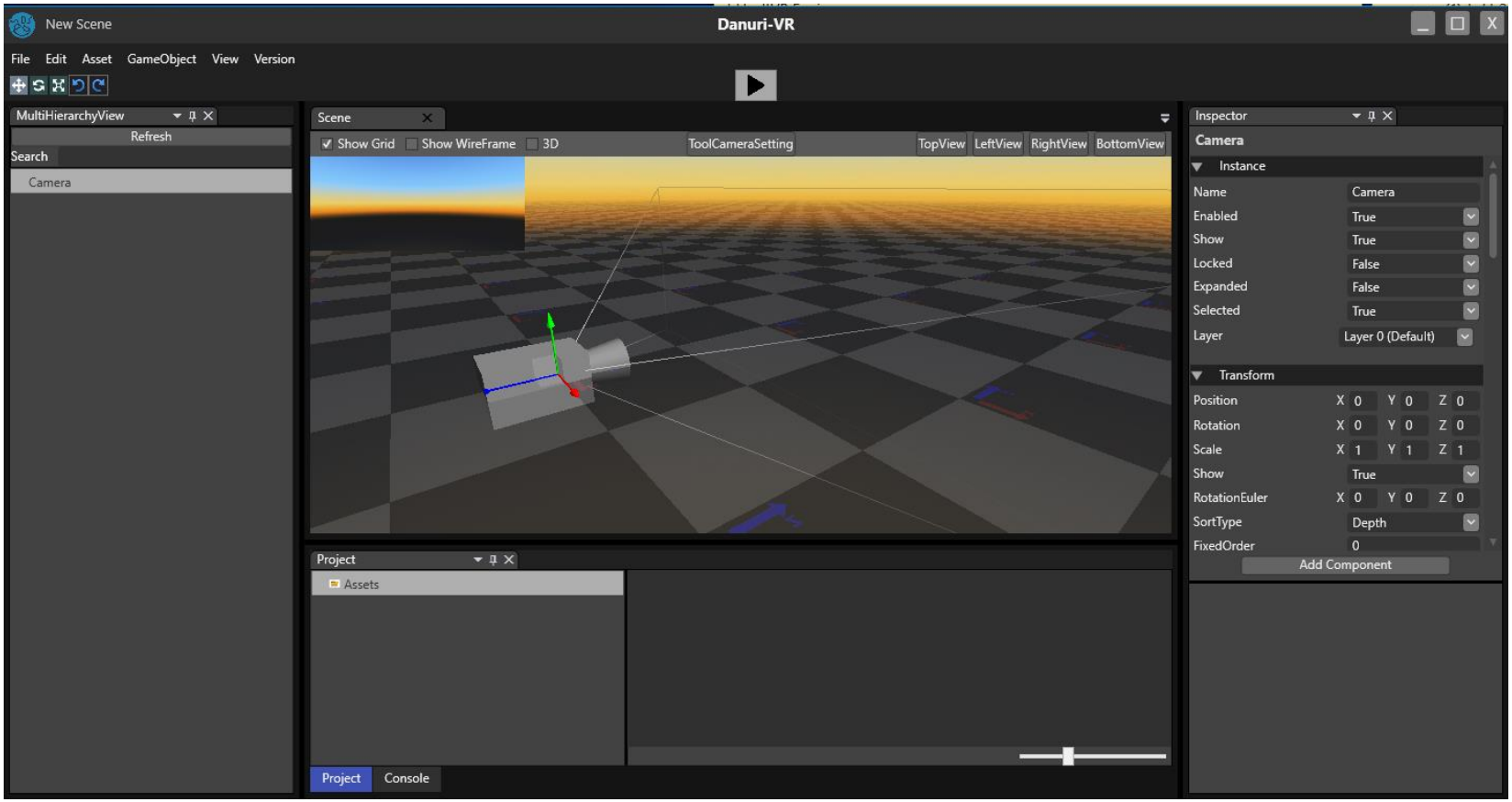
Open Project

- 프로젝트 불러오기
  - FILE메뉴->Open Project(Ctrl+Shift+O)

다누리 VR: Shading & Control

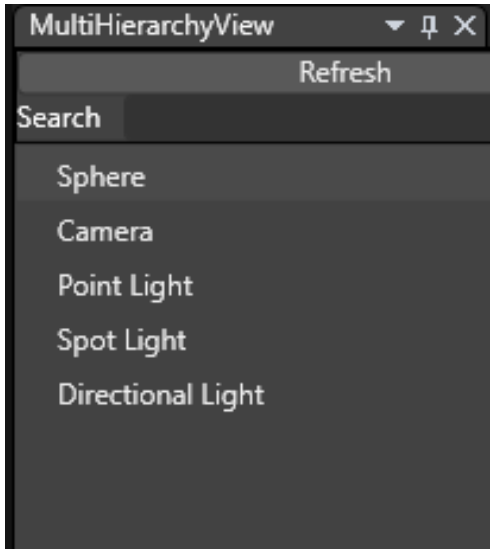
Scene 구성

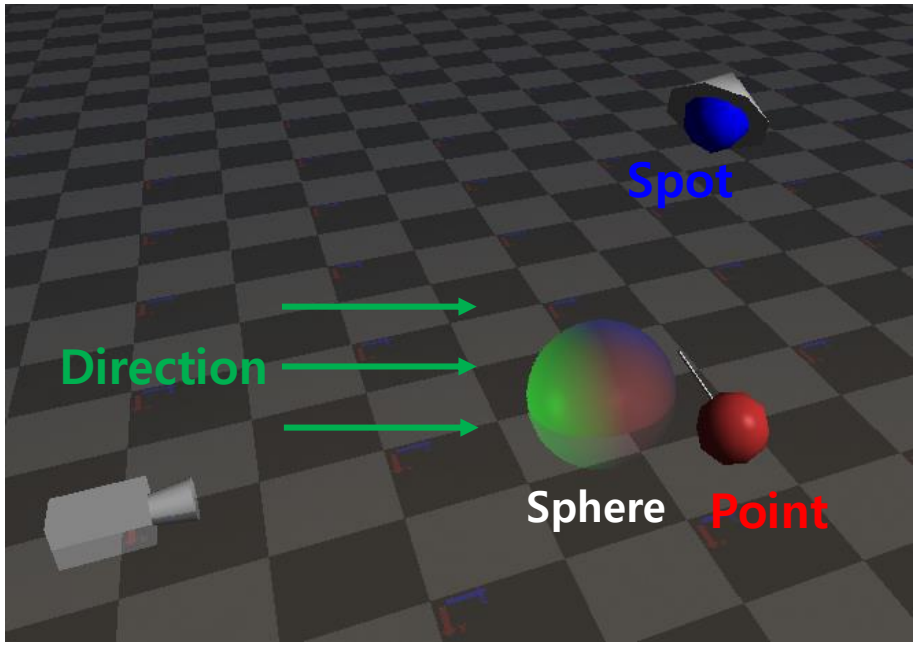
Scene 구성: Camera 생성



- Creation: [Multi Hierarchy View 우클릭] – [Create ContainerSet] –[Camera]
- Setting: [Camera 좌클릭] – [Inspector 에서 위치및 각도 설정]

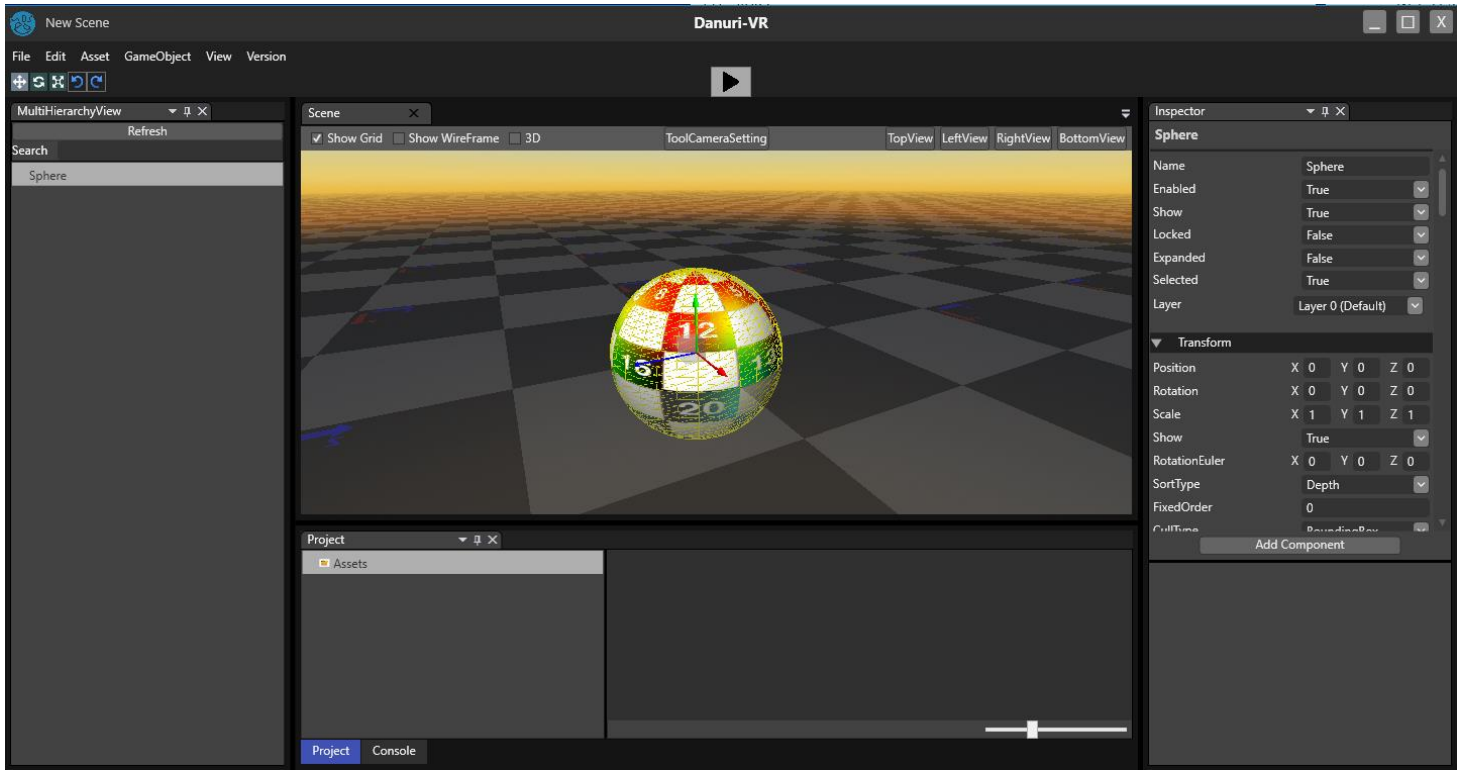
Scene 구성: Light 생성





- Creation: [Multi Hierarchy View 우클릭] – [Create ContainerSet] – [Light] –[Directional/Spot/Point 중 택일]
- Setting: [Inspector 에서 위치및 각도 빛의 색 결정]

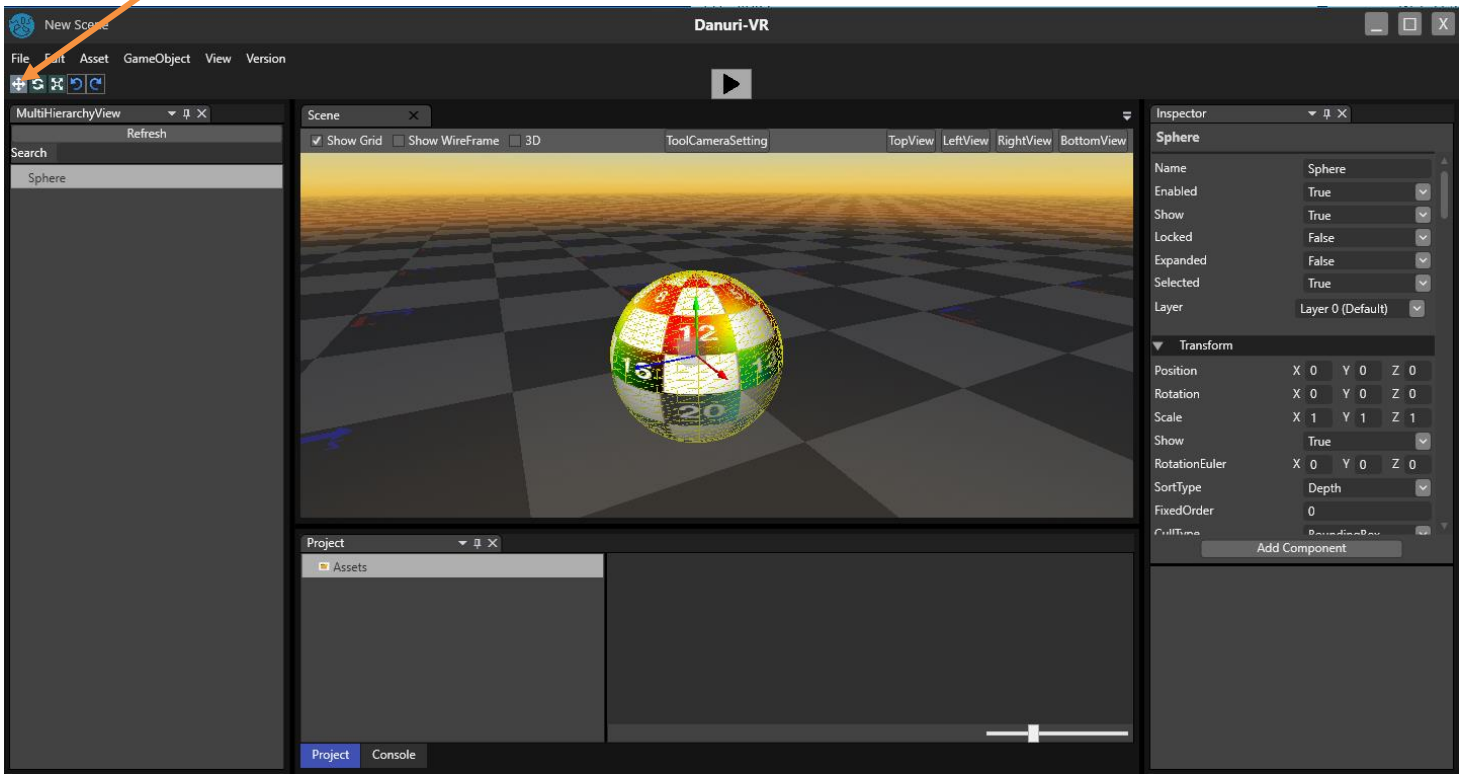
Scene 구성: Object 생성



- Creation: [Multi Hierarchy View 우클릭] – [Create ContainerSet] –[3D Object] – [Sphere]

Scene Control: Translation

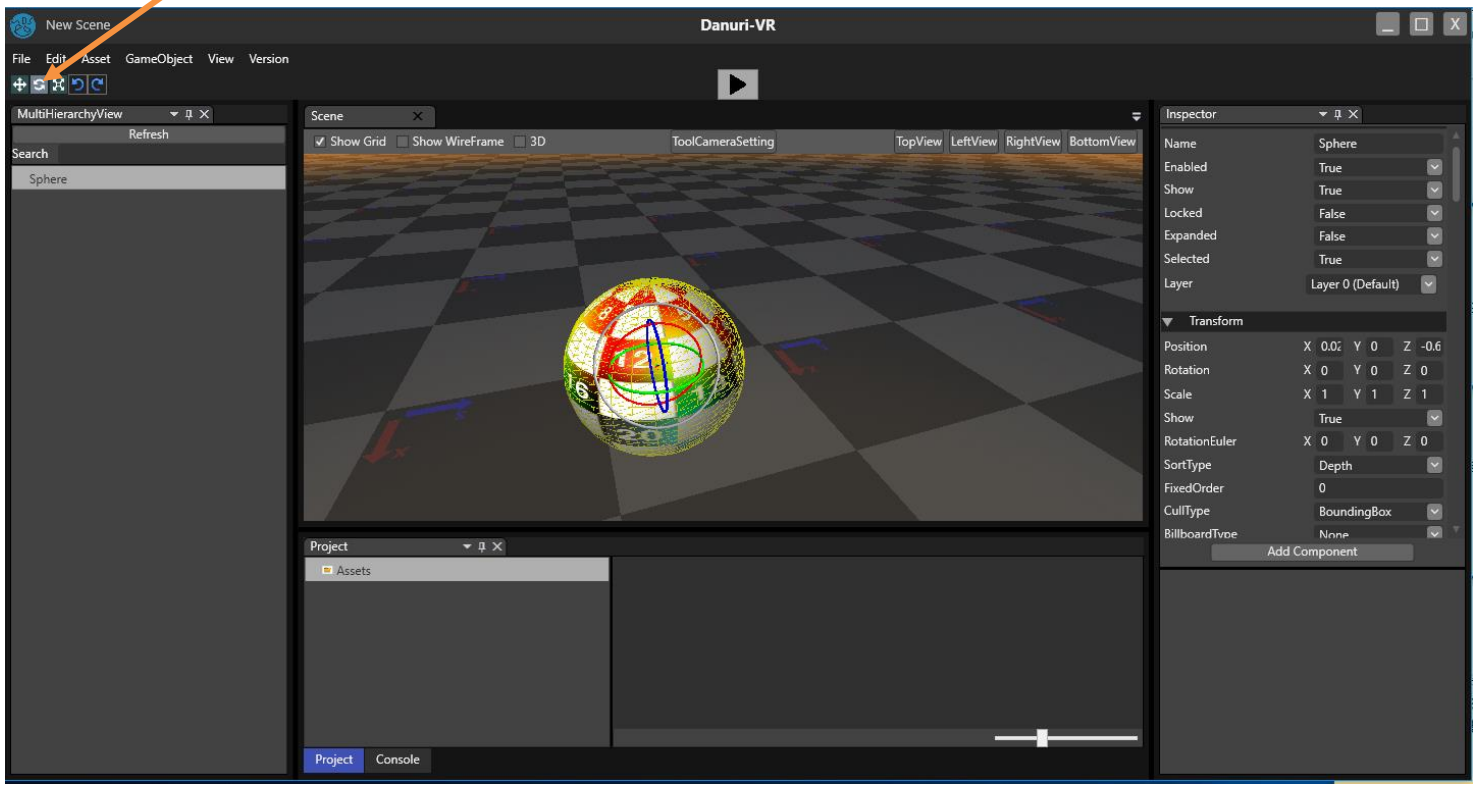
Position key



- [Ctrl+Q] 누른 후, 화살표 잡고 Drag
- 또는 Position key 누름.
- Inspector 창에서 직접 위치 입력

Scene Control: Rotation

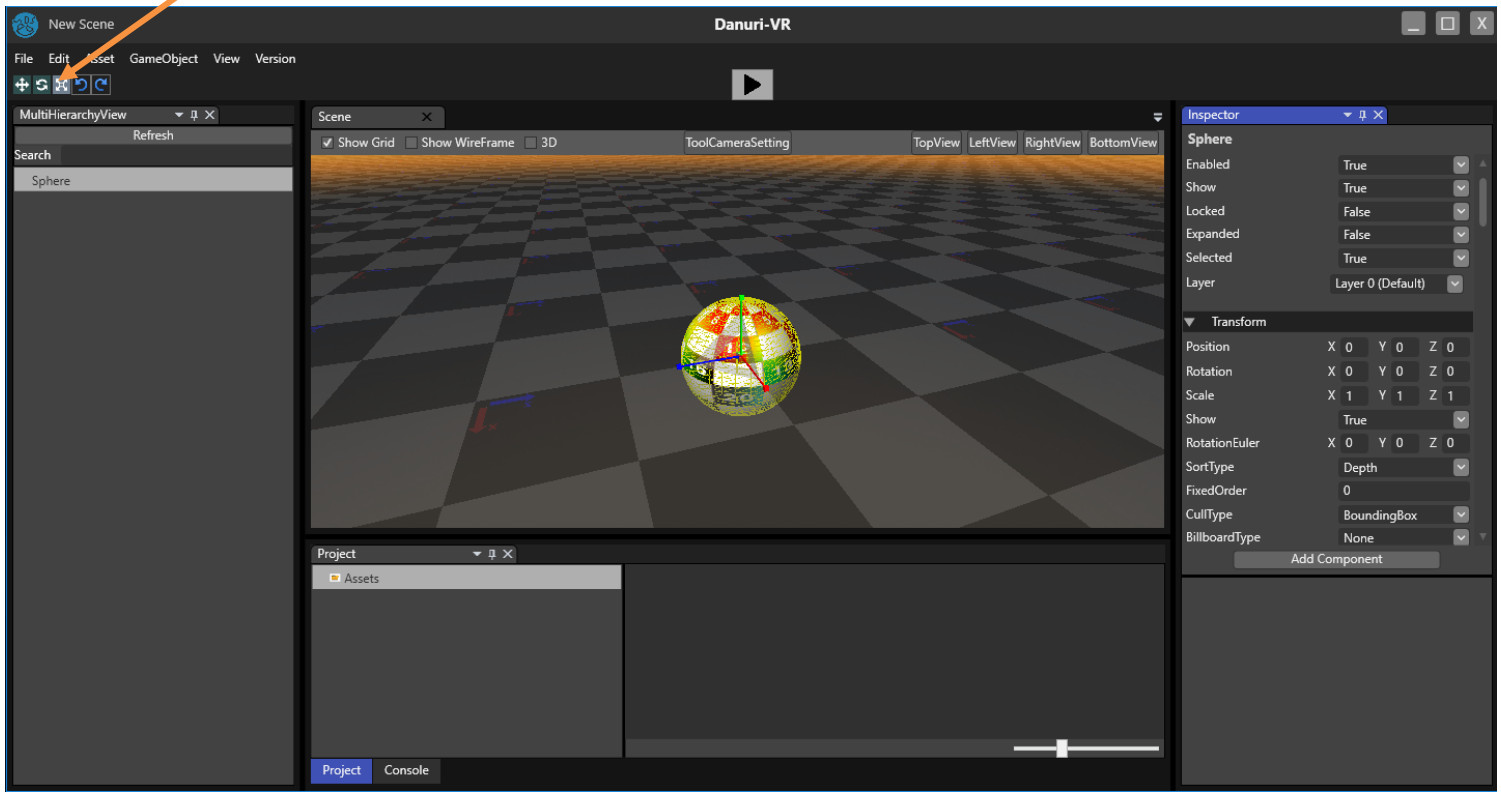
Rotation key



- (Ctrl+W) 누른 후 Object 회전
- 또는 Rotation key 누름
- Inspector 창에서 직접 회전 각도 입력

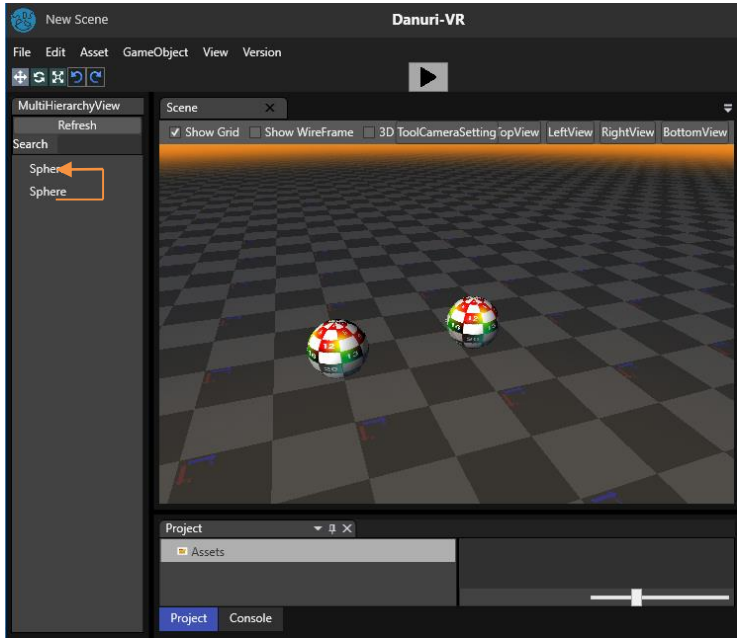
Scene Control: Scaling

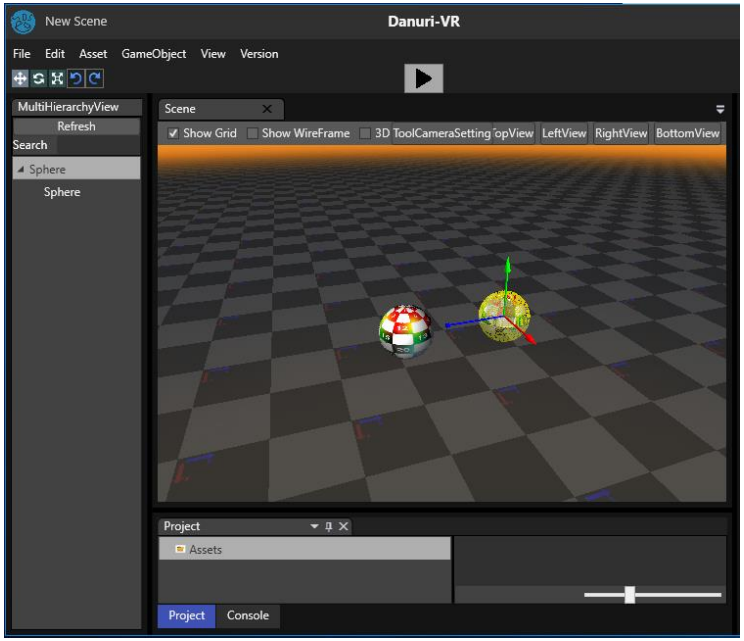
Scale key



- 선택한 객체의 크기를 변경(Ctrl+E)
- Scale key 누름
- Inspector 창에서 직접 팽창 비율 입력

Scene Hierarchical Control



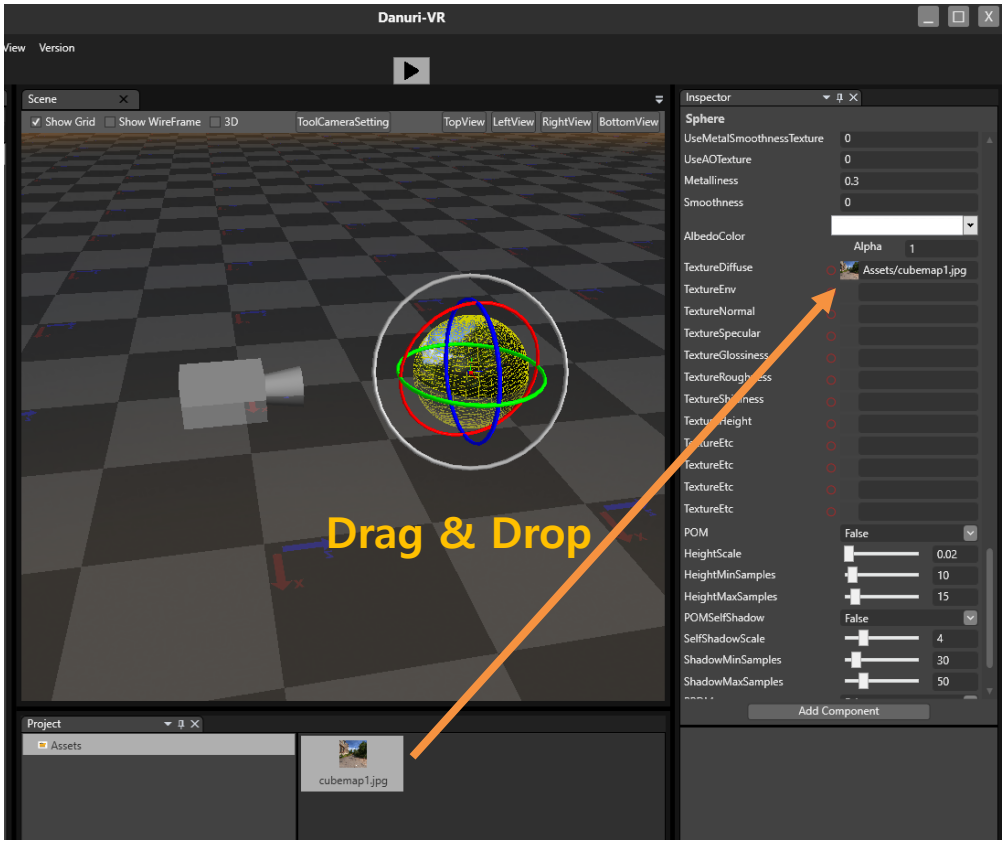


- 오브젝트간 상하관계 적용
  - 하위 오브젝트로 만들 오브젝트를 선택하여 상위 오브젝트로 만들 객체로 드래그하면 오브젝트간 상하관계를 적용할 수 있음
- 상위 오브젝트 위치 변경 시 하위 오브젝트가 같이 움직임을 확인.

Shading & Control

Shading

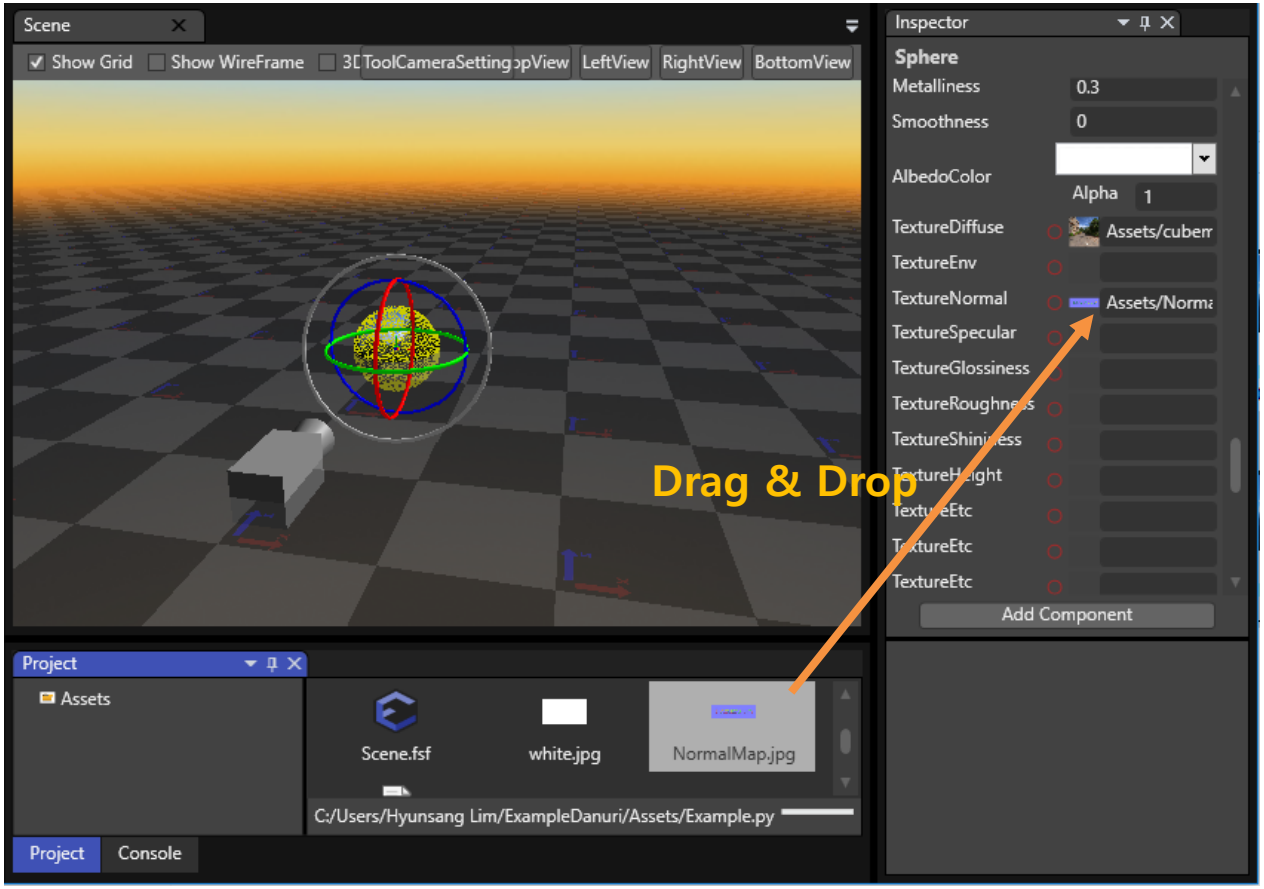
Texture Mapping



- 텍스쳐 선택 버튼을 클릭하면, 에셋폴더 내의 텍스쳐 중 원하는 텍스쳐를 선택할 수 있음

Texture Mapping 결과

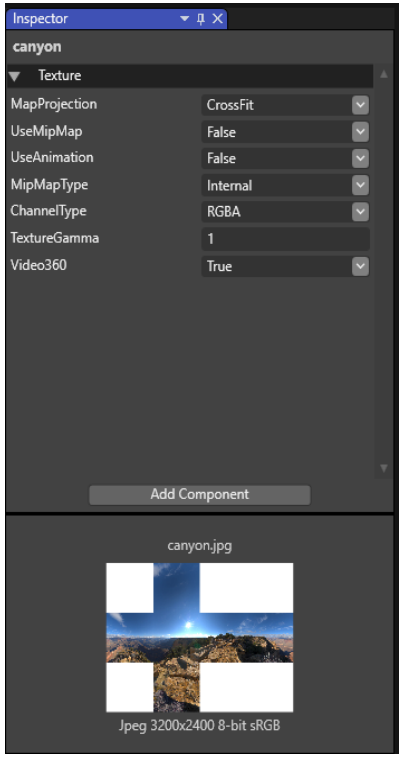
Normal mapping

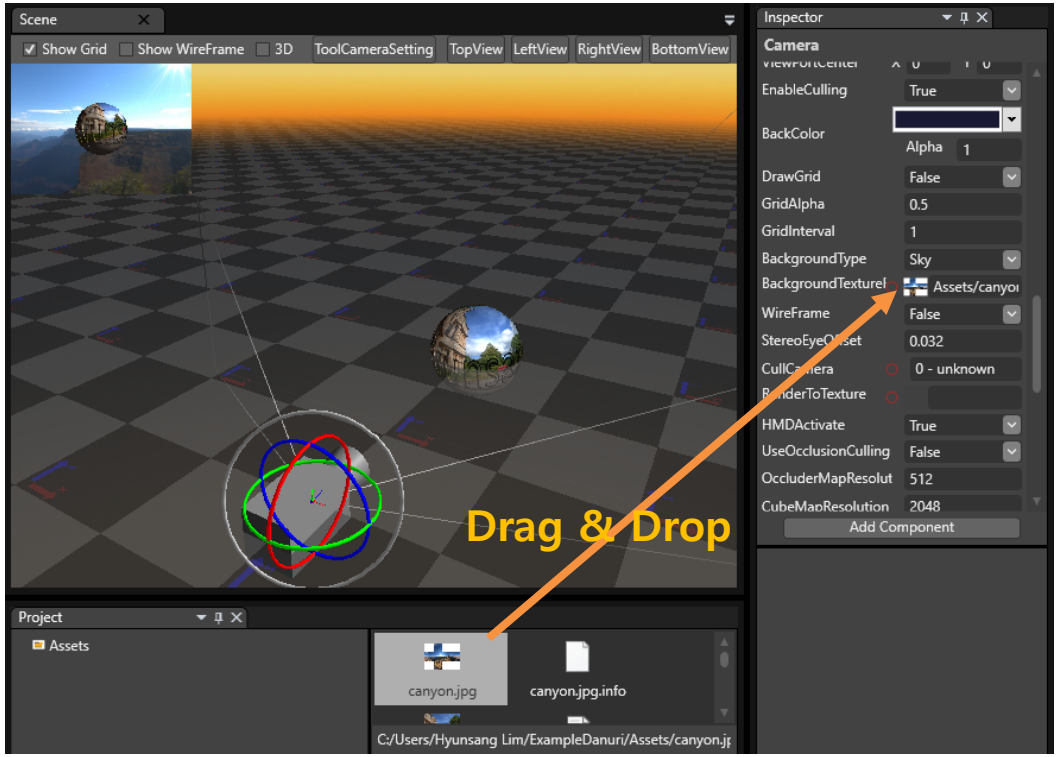


- [Object 좌클릭] - [Inspector의 TextureNormal에 normal map file Drag&Drop]

Normal mapping 결과

Sky Box Mapping

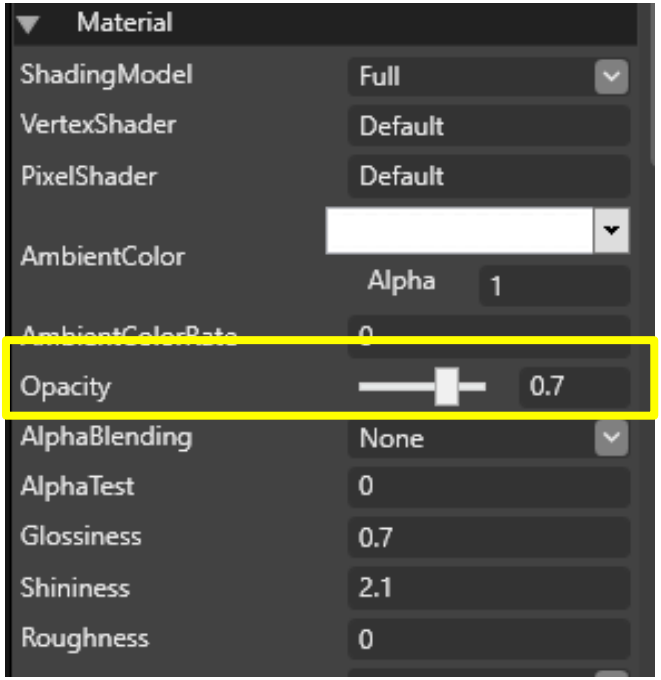




- Texture로 사용할 jpg 파일 MapProjection 을 CrossFit으로 변경
- [Camera 좌클릭] - [Inspector의 BackgroundTextureFile에 사진파일 Drag&Drop]

Sky Box Mapping 결과

Environment mapping: Opacity



- 투명도 조정: Opacity 값 조정

Opacity 결과

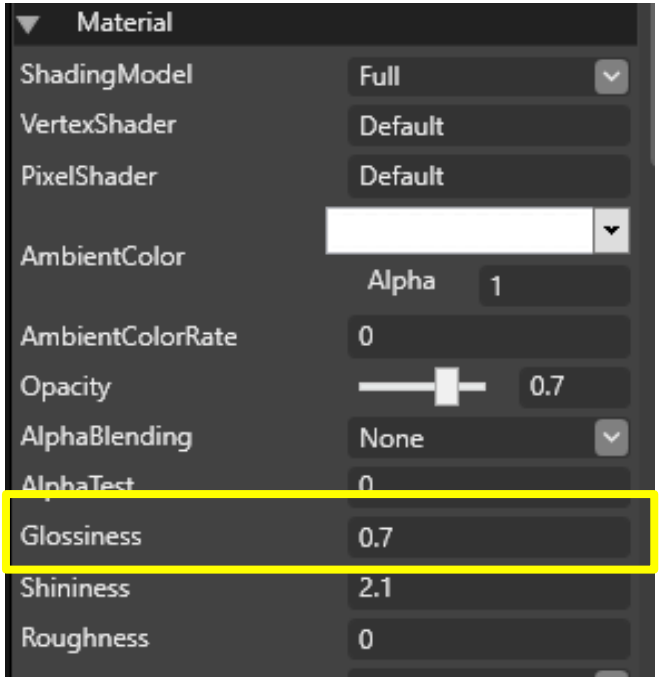
Opacity 1

Opacity 0.7

Opacity 0.3

Opacity 0

Environment mapping: Reflection



- 반사도 조정: Glossiness 값 조정

Reflection 결과

Glossiness 0

Glossiness 0.3

Glossiness 0.7

Glossiness 1.0

Scene Control with Python

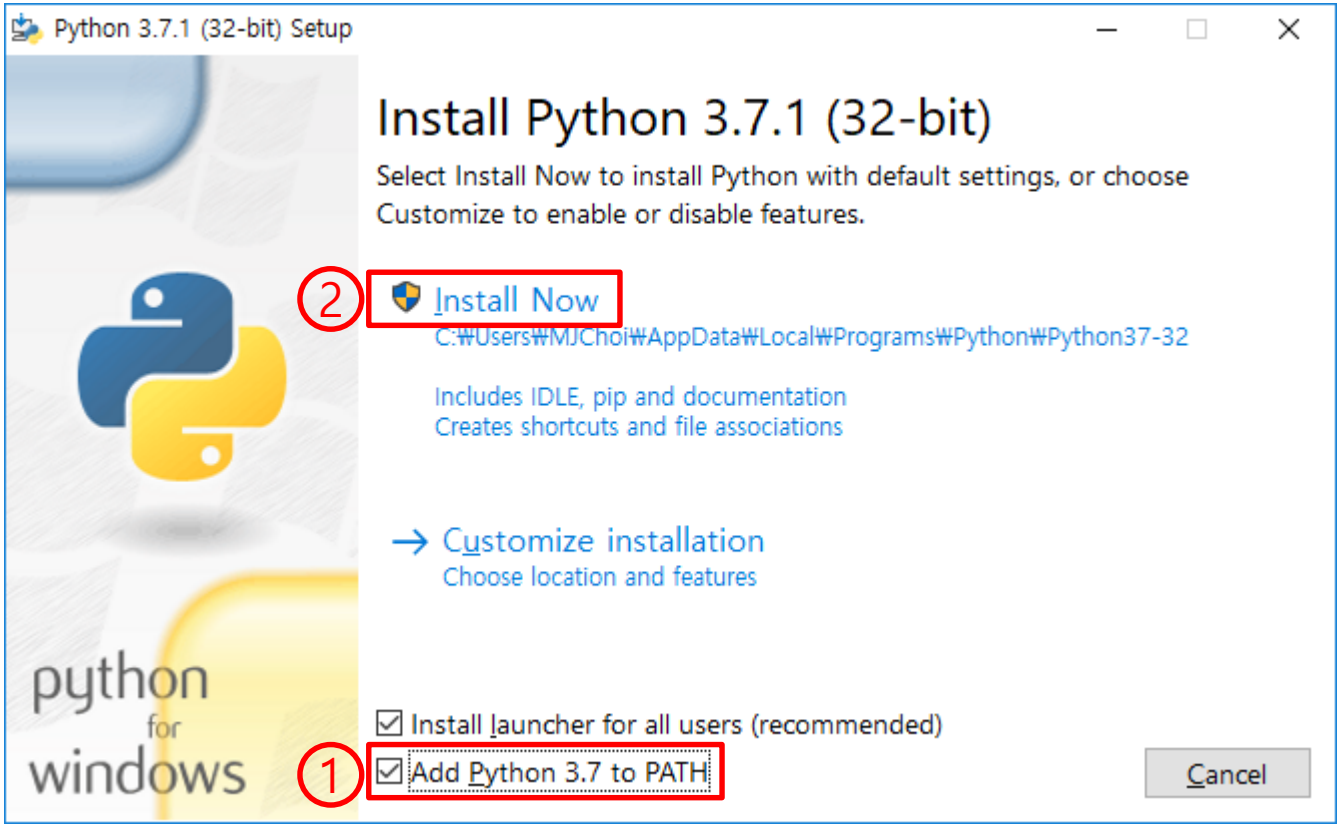
Python설치(1/3)

1. **Python 홈페이지 접속 (https://www.python.org/downloads/)**

2. **프로그램 다운로드**

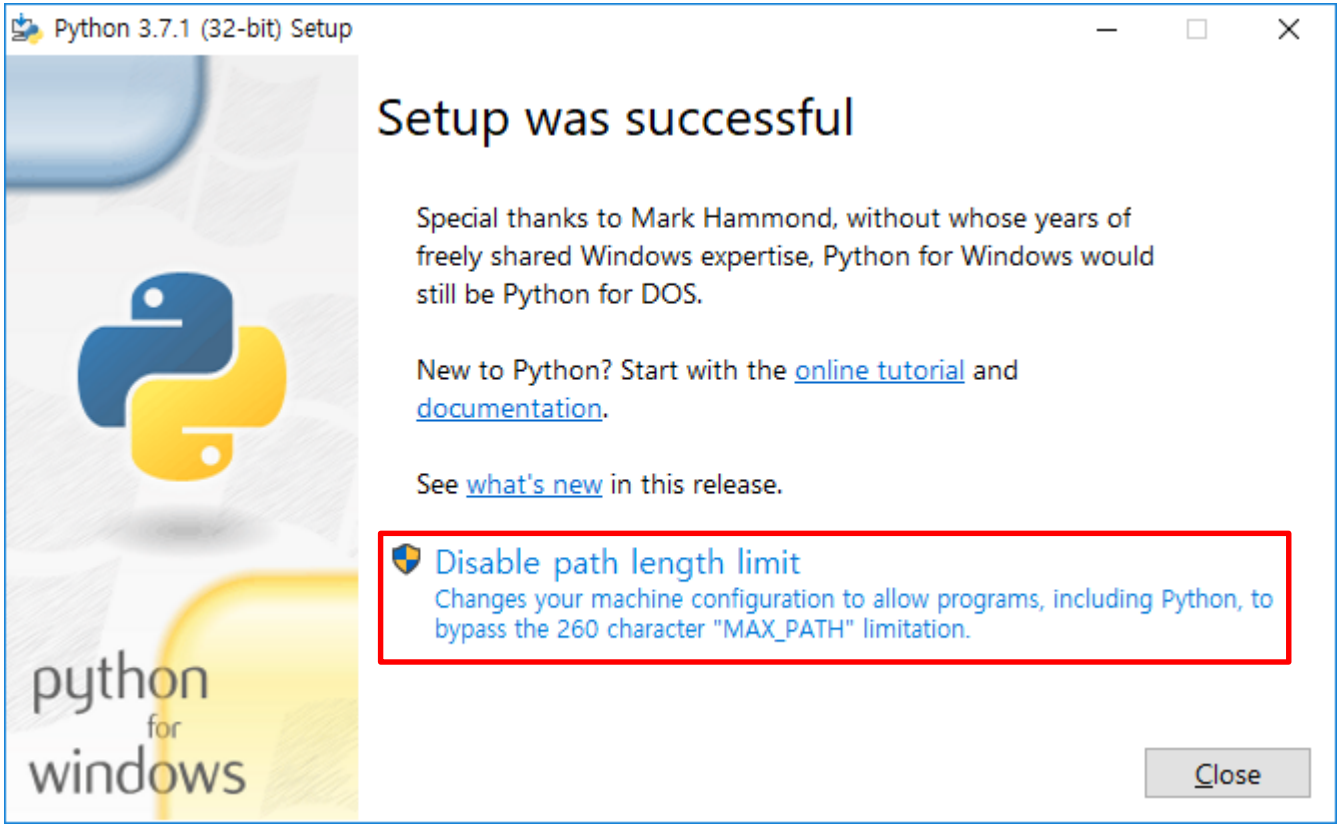
Python설치(2/3)

3. Add Python 3.7 to PATH 체크 후 설치 진행 (Install Now)



Python설치(3/3)

4. Disable path length limit 클릭



Skeleton Python code for Danuri VR

Danuri's own functions and variables are orange color

```
class EX3Main(Actor): #Actor class inheritance( 기존 Danuri서 제공하는 Actor 클래쓰 상속)
    def __init__(self):
        #Initialization
        self.ObjectName = Container(0) #Get Object information
        return
    def OnCreate(self, uid):
        #Constructor
        self.TransformGroup = self.Sphere.FindComponentByType("TransformGroup")#Get Transformation information
        return 0
    def OnDestroy(self):
        return 0
    def Update(self):
       #Updating frame
       self.TransformGroup.ShiftPosition()
       spherePos = self.TransformGroup.GetPosition()
       self.TransformGroup.Rotate()
       self.TransformGroup.SetScale()

       return

def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):

        if (msg == "KeyDown"):
            if( number == Hexa code):
                #Keyboard callback command
        return;
```

Translation Transformation:

```
class EX3Main(Actor):
    def __init__(self):
        self.Sphere = Container(0) #Get Object information
        return
    def OnCreate(self, uid):
        self.TransformGroup = self.Sphere.FindComponentByType("TransformGroup")#Get Transformation information
        return 0
    def OnDestroy(self):
        return 0
    def OnEnable(self):
        return 0
    def OnDisable(self):
        return 0
    def Update(self):
        return

def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):
        print(msg)

        if (msg == "KeyDown"):
            if( number == 0x51): #"Q"
                self.TransformGroup.ShiftPosition(Math3d.Vector3(0.1,0.0,0.0))

            if( number == 0x57): #"W"
                self.TransformGroup.ShiftPosition(Math3d.Vector3(0.1,0.0,0.0))
        return;
```

Translation 결과

Rotation Transformation

```
class EX3Main(Actor):
    def __init__(self):
        self.Sphere = Container(0) #Get Object information
        return
    def OnCreate(self, uid):
        self.TransformGroup = self. Sphere.FindComponentByType("TransformGroup")#Get Transformation information
        return 0
    def OnDestroy(self):
        return 0
    def OnEnable(self):
        return 0
    def OnDisable(self):
        return 0
    def Update(self):
        return

def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):
        print(msg)

        if (msg == "KeyDown"):
            if( number == 0x51): #"Q"
                spherePos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(2,0, spherePos)

            if( number == 0x57): #"W"
                spherePos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(-2,0, spherePos)

        return;
```

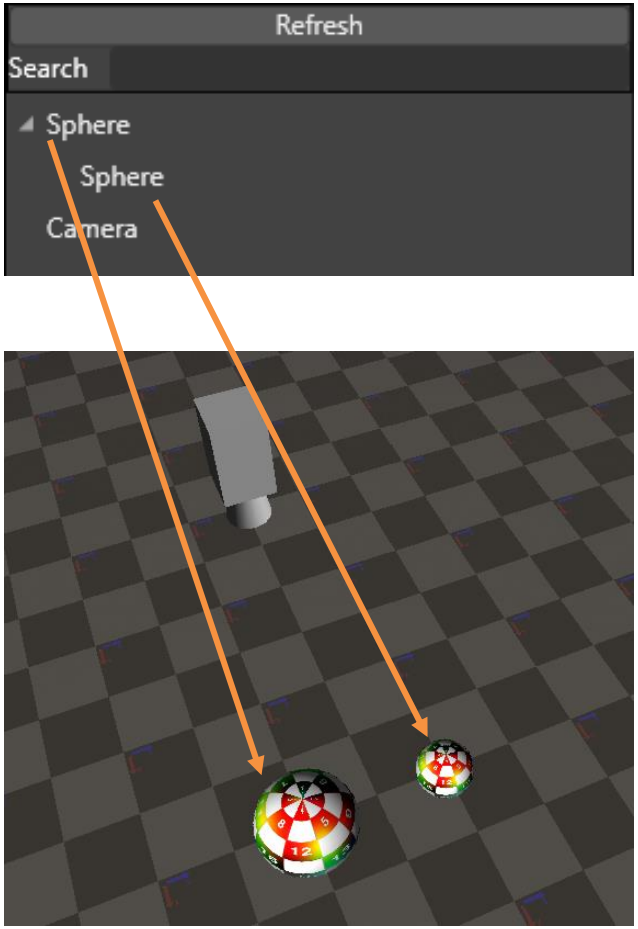
Rotation 결과

Scaling Transformation:

```
class EX3Main(Actor):
    scalefactor = 1.0
    def __init__(self):
        self.Sphere = Container(0) #Get Object information
        return
    def OnCreate(self, uid):
        self.TransformGroup = self. Sphere.FindComponentByType("TransformGroup")#Get Transformation information
        return 0
    def OnDestroy(self):
        return 0
    def OnEnable(self):
        return 0
    def OnDisable(self):
        return 0
    def Update(self):
        return
def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):
        print(msg)

        if (msg == "KeyDown"):
            if( number == 0x51): #"Q"
                self.scalefactor = 1.1*self.scalefactor
                self.TransformGroup.SetScale(Math3d.Vector3(self.scalefactor,self.scalefactor,self.scalefactor))
            if( number == 0x57): #"W"
                self.scalefactor = 0.91*self.scalefactor
                self.TransformGroup.SetScale(Math3d.Vector3(self.scalefactor,self.scalefactor,self.scalefactor))
        return;
```

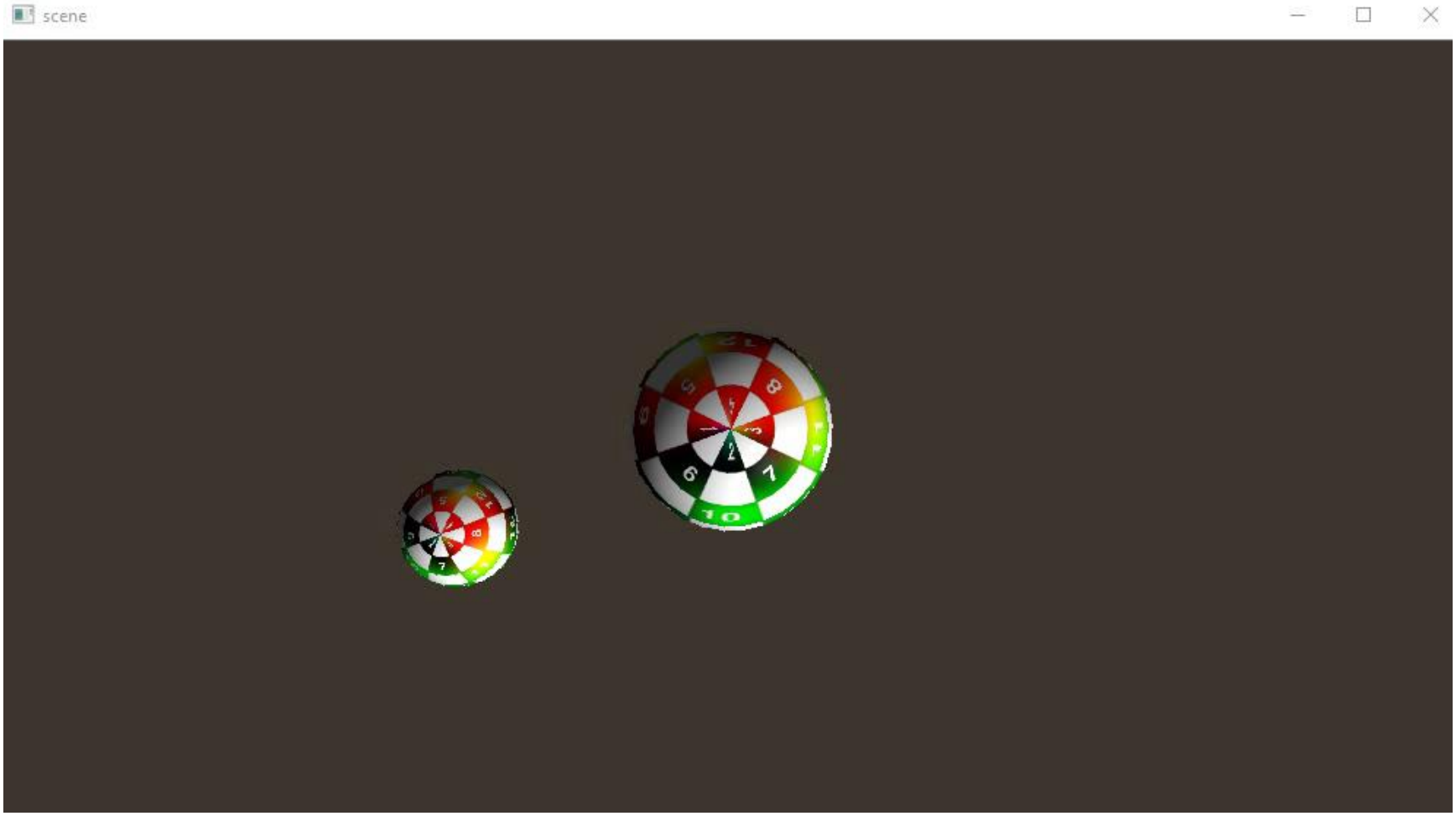
Scaling 결과

Hierarchical Transformation Control

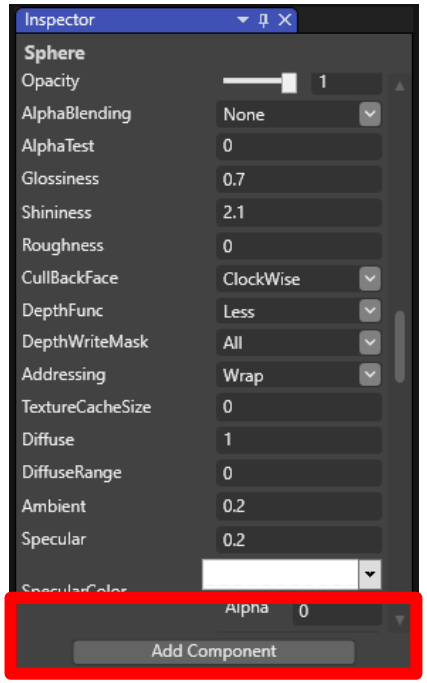


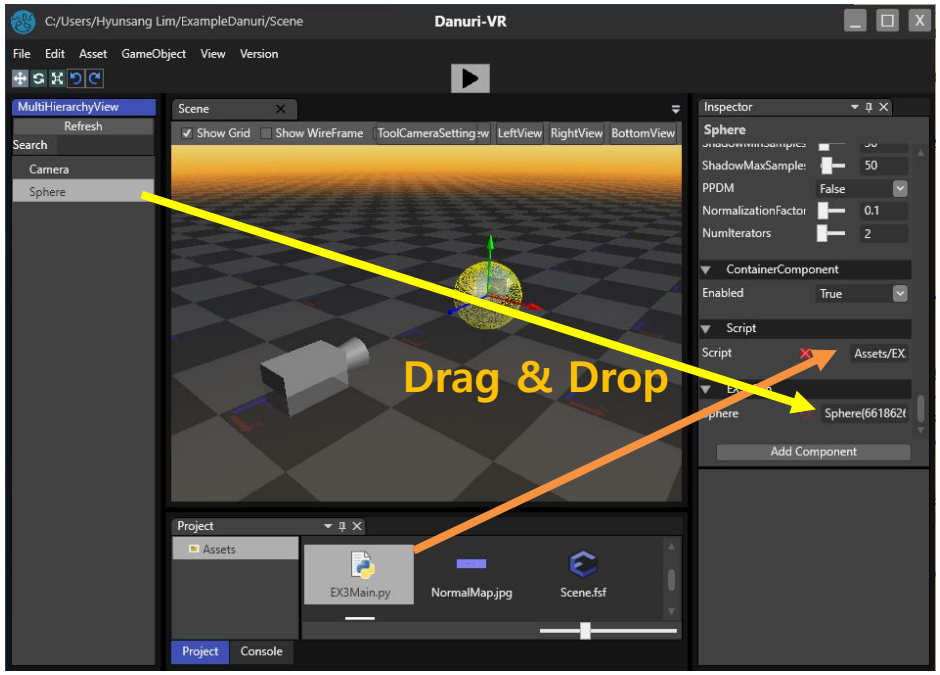

```
//Sphere1
def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):
        print(msg)

        if (msg == "KeyDown"):
            if( number == 0x51): #"Q"
                fbxPos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(2,0,fbxPos)

            if( number == 0x57): #"W"
                fbxPos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(-2,0,fbxPos)

        return;


//Sphere2
def OnMessage(self, msg, number, Vector4_lparm, Vector4_wparam):
        print(msg)

        if (msg == "KeyDown"):
            if( number == 0x45): #“E"
                fbxPos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(2,0,fbxPos)

            if( number == 0x52): #“R"
                fbxPos = self.TransformGroup.GetPosition()
                self.TransformGroup.Rotate(-2,0,fbxPos)

        return;
```

Hierarchical control 결과



Importing Python code



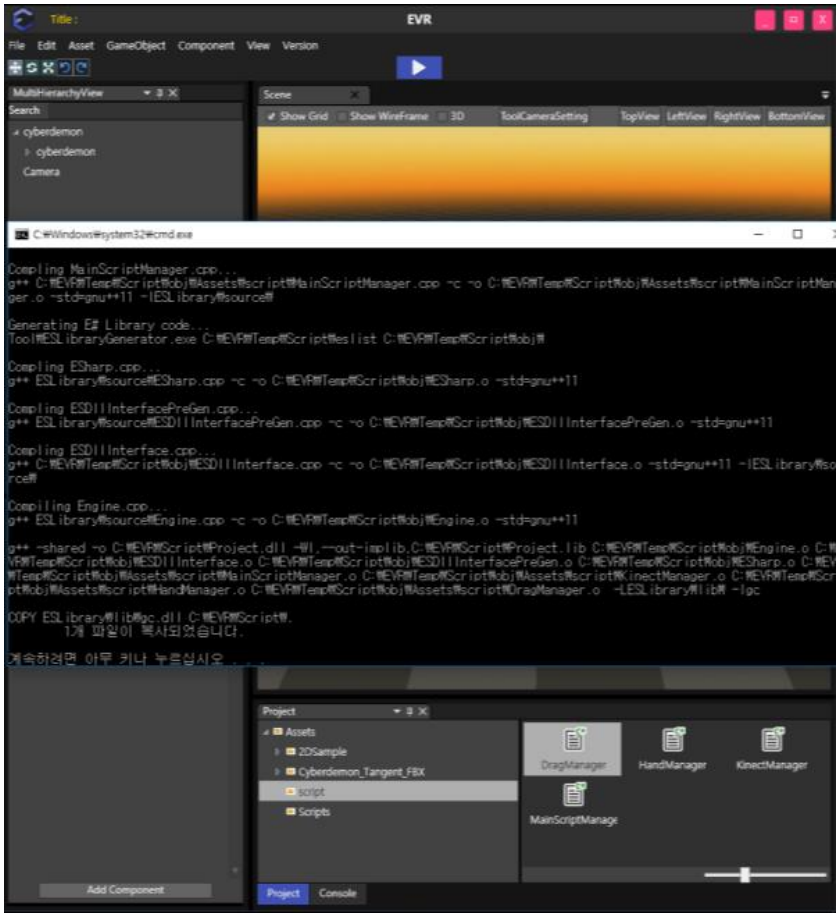


[Object 좌클릭] - [Inspector의 Add Component] – [script] –[script Component]

[***.py 프로그램 Script에 Drag&Drop] – [Sphere에 Sphere Drag&Drop]

Build

- Script가 있는 프로젝트는 File->Build(Shift+B)를 통해 빌드를 할 수 있다.



Result Example