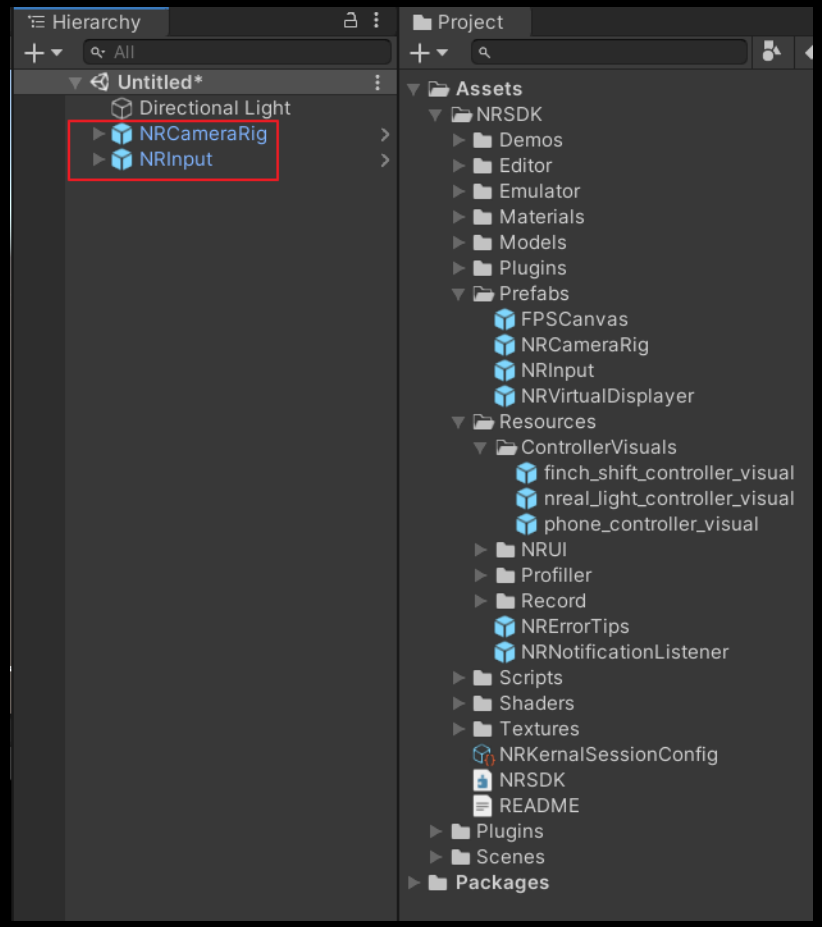
2021.06.09

Nreal R&D

작성자 : 김은규

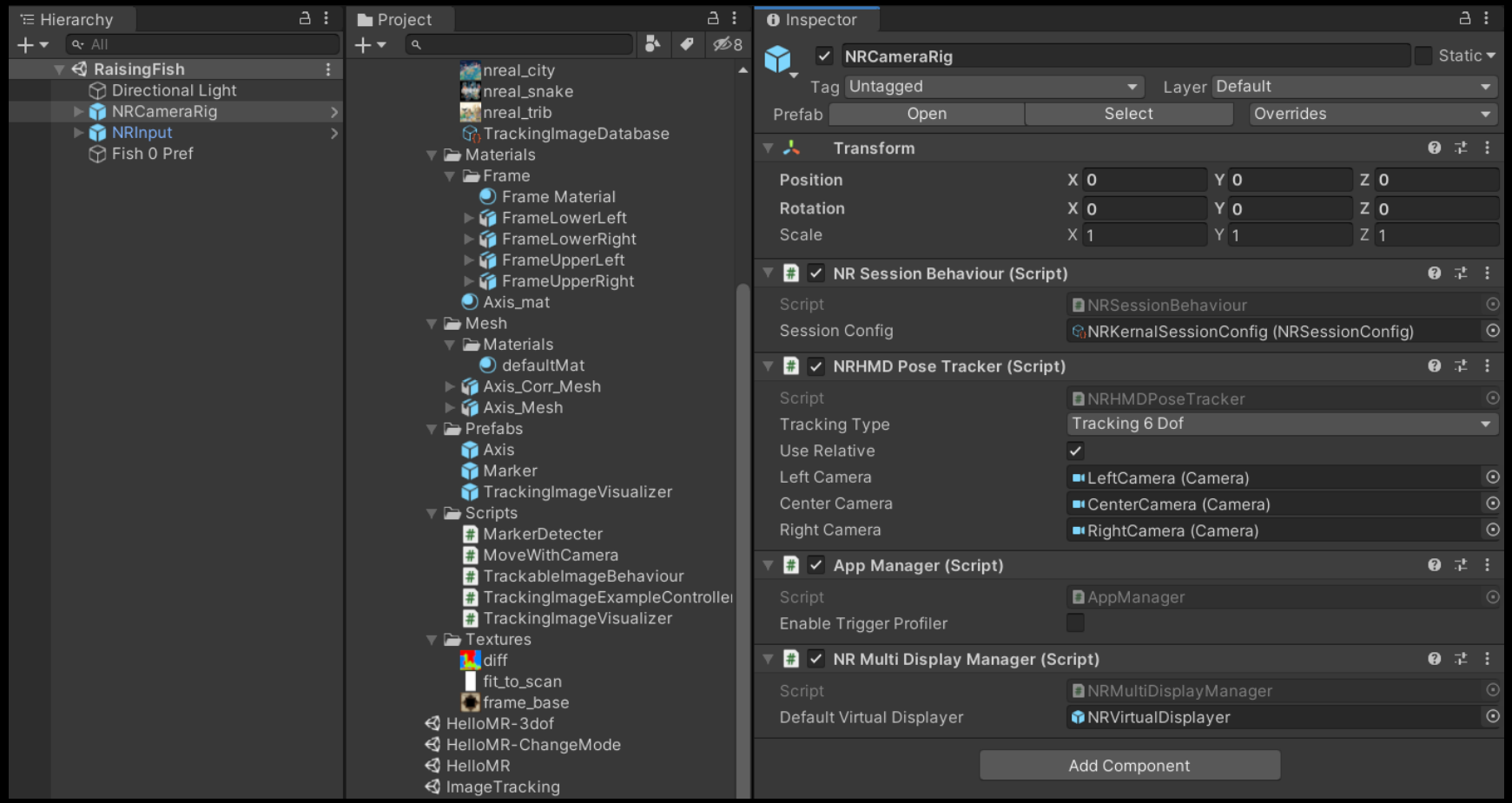
목차

1. Nreal Demo Project



1. New Scene을 합니다.
2. 기존 Main Camera를 제거 합니다.
3. CameraNRCameraRig를 추가합니다.
4. NRInput을 추가합니다.

NRCameraRig

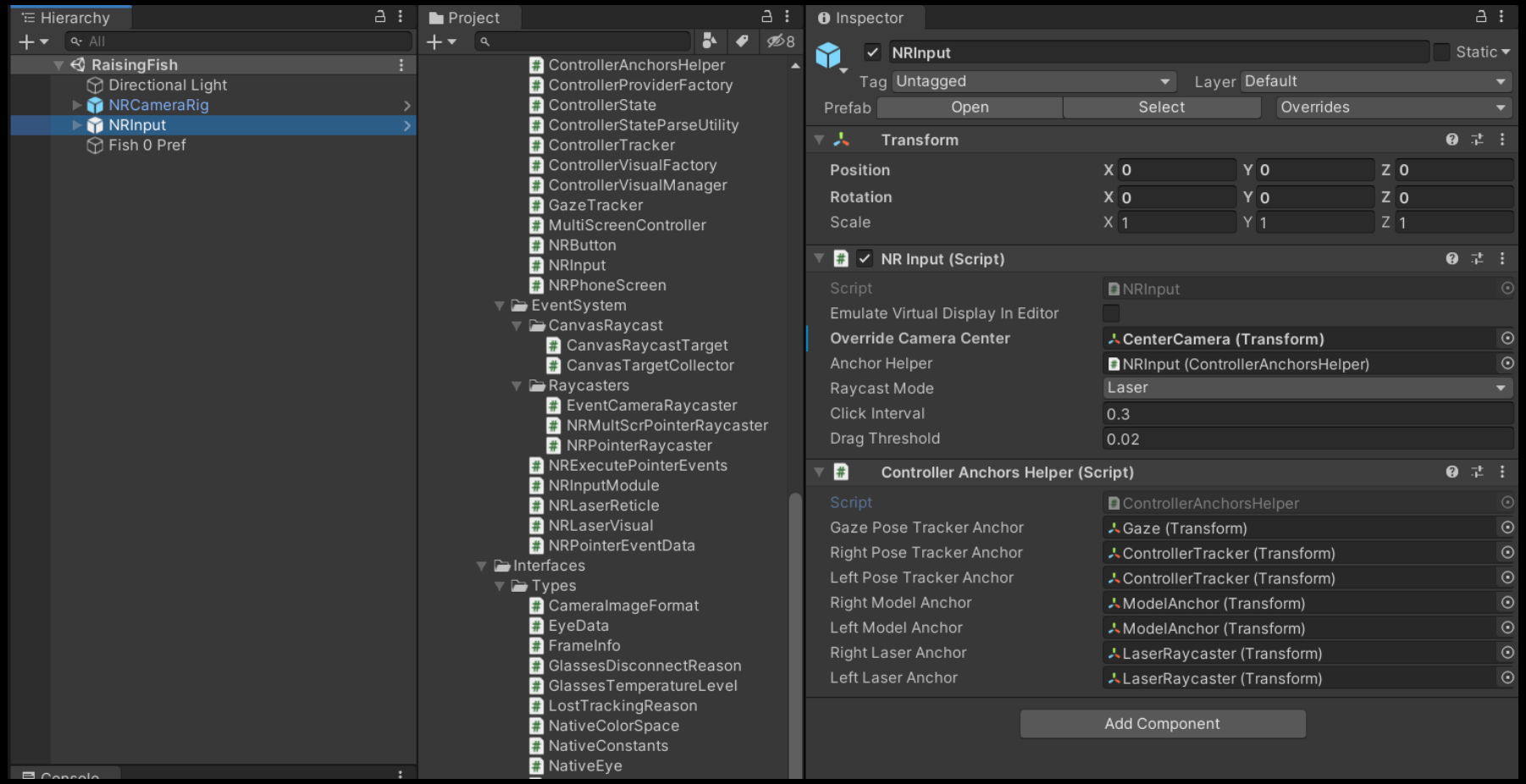


NRCameraRig

- NRSessionBehaviour
- AR 시스템 상태를 작동하고 애플리케이션 계층의 세션 라이프사이클을 처리합니다.

- NRHMDPoseTracker
- HMPoseTracker는 pose tracker의 정보를 업데이트합니다. 이 구성 요소는 카메라 매개 변수를 초기화하고 장치 상태를 업데이트하는 데 사용됩니다. 또한 응용 프로그램이 이 구성 요소를 통해 추적 유형을 변경할 수 있습니다.

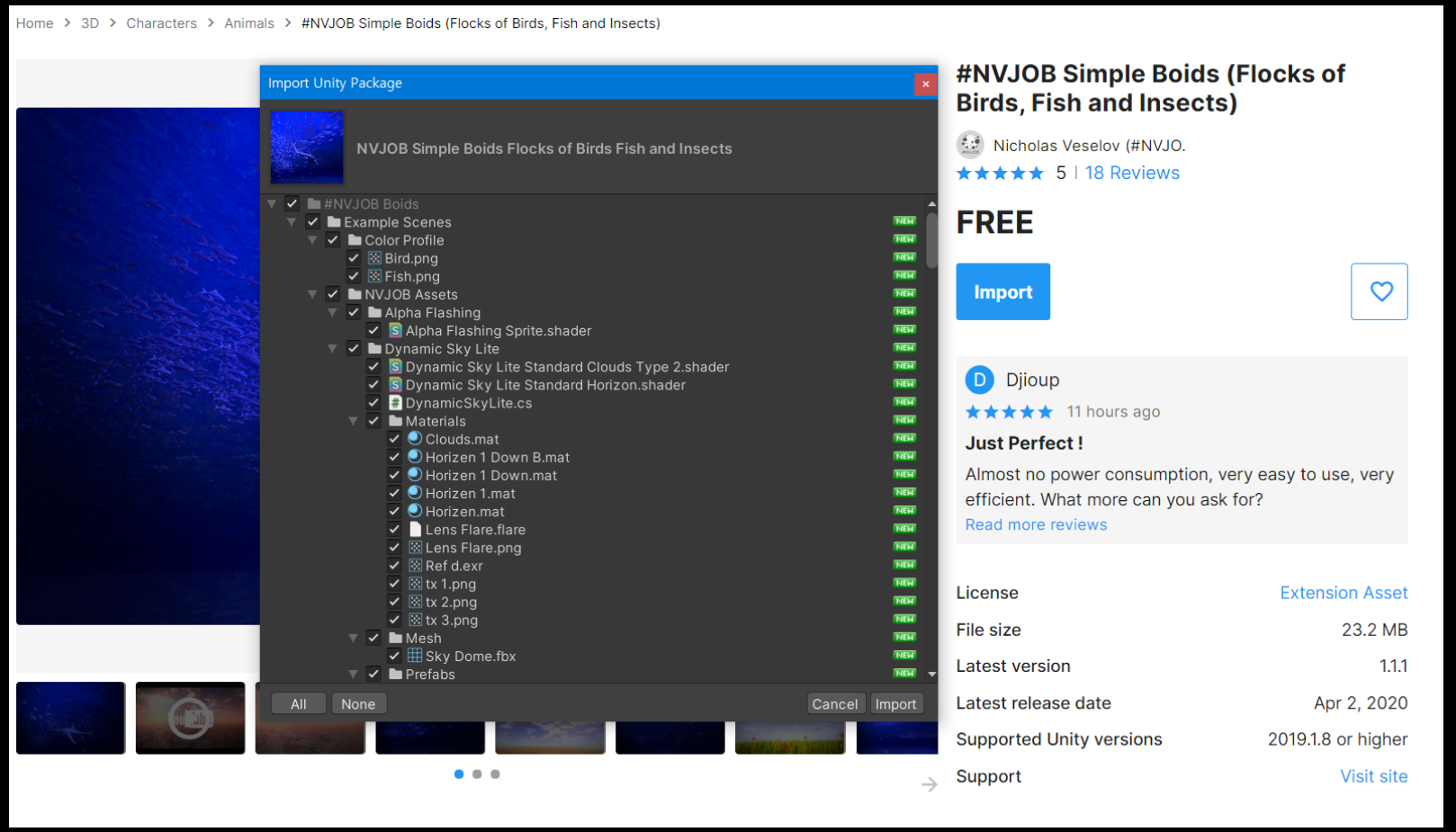
NRInput


Hierarchy
All
RaisingFish
Directional Light
NRCameraRig
NRInput
Fish 0 Pref
Project
ControllerAnchorsHelper
ControllerProviderFactory
ControllerState
ControllerStateParseUtility
ControllerTracker
ControllerVisualFactory
ControllerVisualManager
GazeTracker
MultiScreenController
NRButton
NRInput
NRPhoneScreen
EventSystem
CanvasRaycast
CanvasRaycastTarget
CanvasTargetCollector
Raycasters
EventCameraRaycaster
NRMultScrPointerRaycaster
NRPointerRaycaster
NRExecutePointerEvents
NRInputModule
NRLaserReticle
NRLaserVisual
NRPointerEventData
Interfaces
Types
CameraImageFormat
EyeData
FrameInfo
GlassesDisconnectReason
GlassesTemperatureLevel
LostTrackingReason
NativeColorSpace
NativeConstants
NativeEye
Inspector
NRInput
Static
Tag Untagged
Layer Default
Prefab Open Select Overrides
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
NR Input (Script)
Script NRInput
Emulate Virtual Display In Editor
Override Camera Center CenterCamera (Transform)
Anchor Helper NRInput (ControllerAnchorsHelper)
Raycast Mode Laser
Click Interval 0.3
Drag Threshold 0.02
Controller Anchors Helper (Script)
Script ControllerAnchorsHelper
Gaze Pose Tracker Anchor Gaze (Transform)
Right Pose Tracker Anchor ControllerTracker (Transform)
Left Pose Tracker Anchor ControllerTracker (Transform)
Right Model Anchor ModelAnchor (Transform)
Left Model Anchor ModelAnchor (Transform)
Right Laser Anchor LaserRaycaster (Transform)
Left Laser Anchor LaserRaycaster (Transform)
Add Component

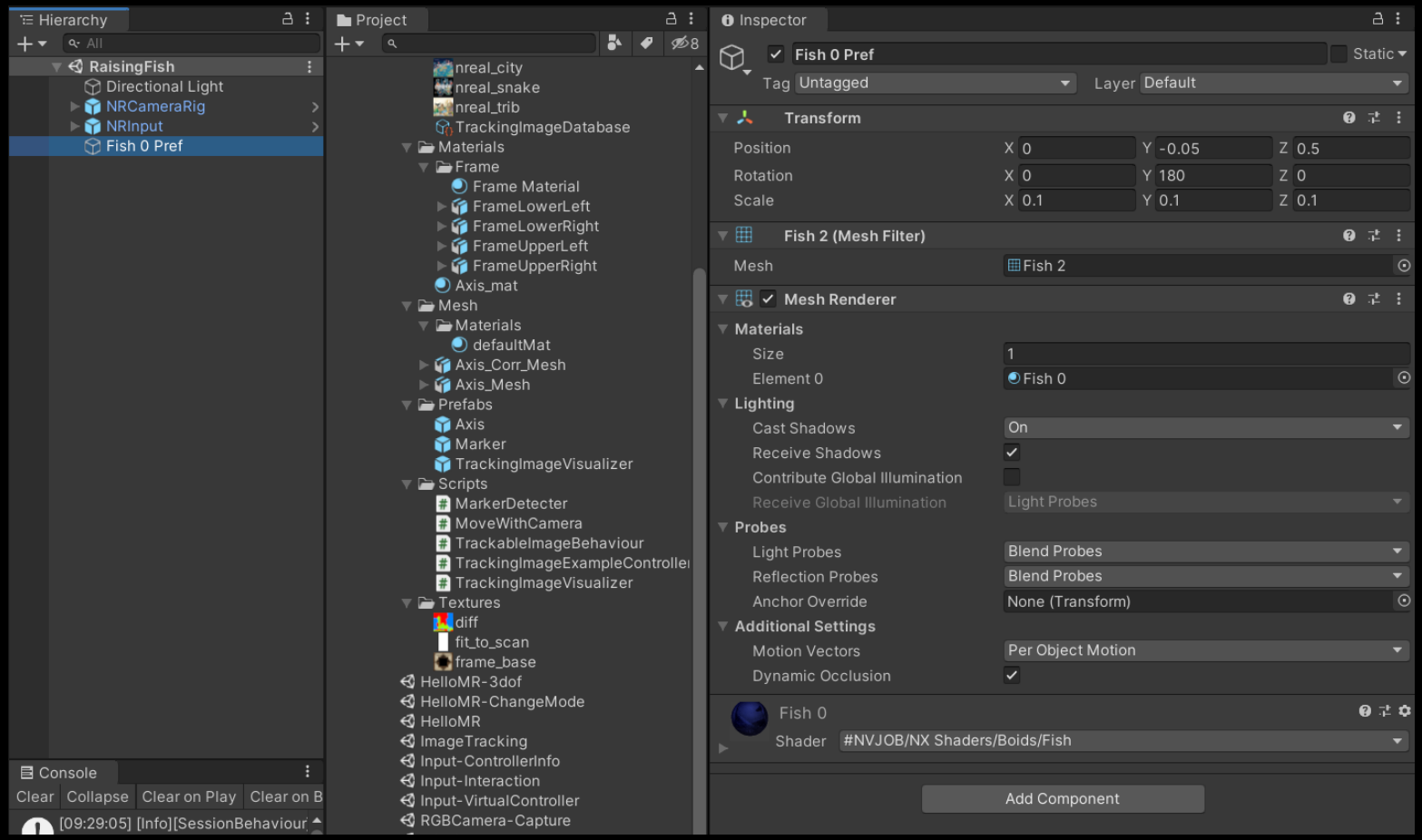
NRInput

- 컨트롤러 상태를 가져오거나 컨트롤러 상태를 업데이트하는 등의 컨트롤러 관련 작업을 처리하는 메인 클래스는 이 클래스를 통해 애플리케이션이 사용자 지정이 될 수 있는 컨트롤러 프로바이더를 생성한 다음 컨트롤러 공급자 자신이 컨트롤러 상태를 업데이트하는 방법을 정의하여 모든 프레임 NRnut가 올바른 상태를 가져올 수 있도록 합니다.하나의 컨트롤러 공급자에 대해 최대 두 개의 상태가 있습니다.

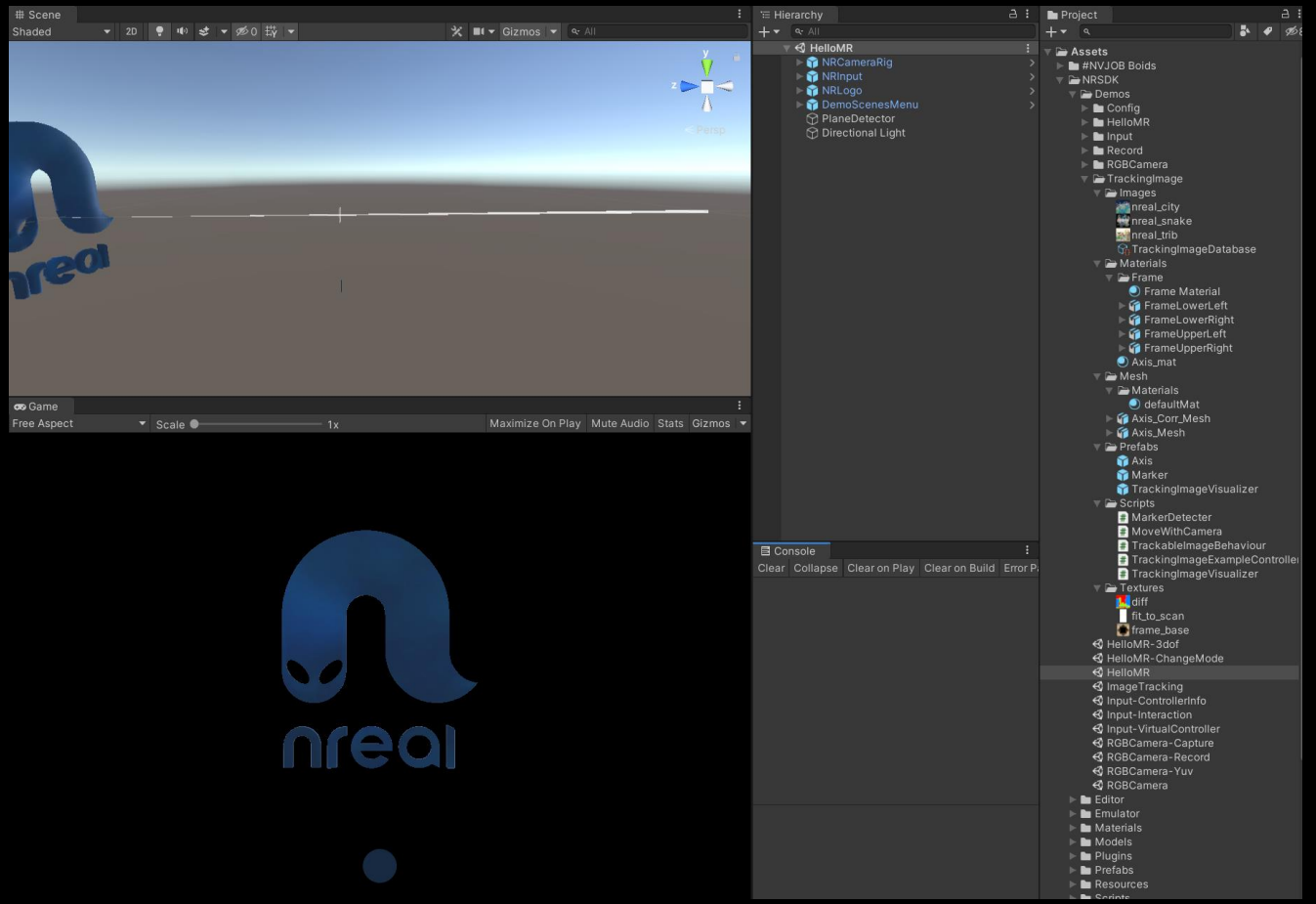
Asset 추가


Home > 3D > Characters > Animals > #NVJOB Simple Boids (Flocks of Birds, Fish and Insects)
Import Unity Package
NVJOB Simple Boids Flocks of Birds Fish and Insects
#NVJOB Boids
Example Scenes
Color Profile
Bird.png
Fish.png
NVJOB Assets
Alpha Flashing
Alpha Flashing Sprite.shader
Dynamic Sky Lite
Dynamic Sky Lite Standard Clouds Type 2.shader
Dynamic Sky Lite Standard Horizon.shader
DynamicSkyLite.cs
Materials
Clouds.mat
Horizen 1 Down B.mat
Horizen 1 Down.mat
Horizen 1.mat
Horizen.mat
Lens Flare.flare
Lens Flare.png
Ref d.exr
tx 1.png
tx 2.png
tx 3.png
Mesh
Sky Dome.fbx
Prefabs
All
None
Cancel
Import
#NVJOB Simple Boids (Flocks of Birds, Fish and Insects)
Nicholas Veselov (#NVJO.
5 | 18 Reviews
FREE
Import
Djioup
11 hours ago
Just Perfect !
Almost no power consumption, very easy to use, very efficient. What more can you ask for?
Read more reviews
License
Extension Asset
File size
23.2 MB
Latest version
1.1.1
Latest release date
Apr 2, 2020
Supported Unity versions
2019.1.8 or higher
Support
Visit site

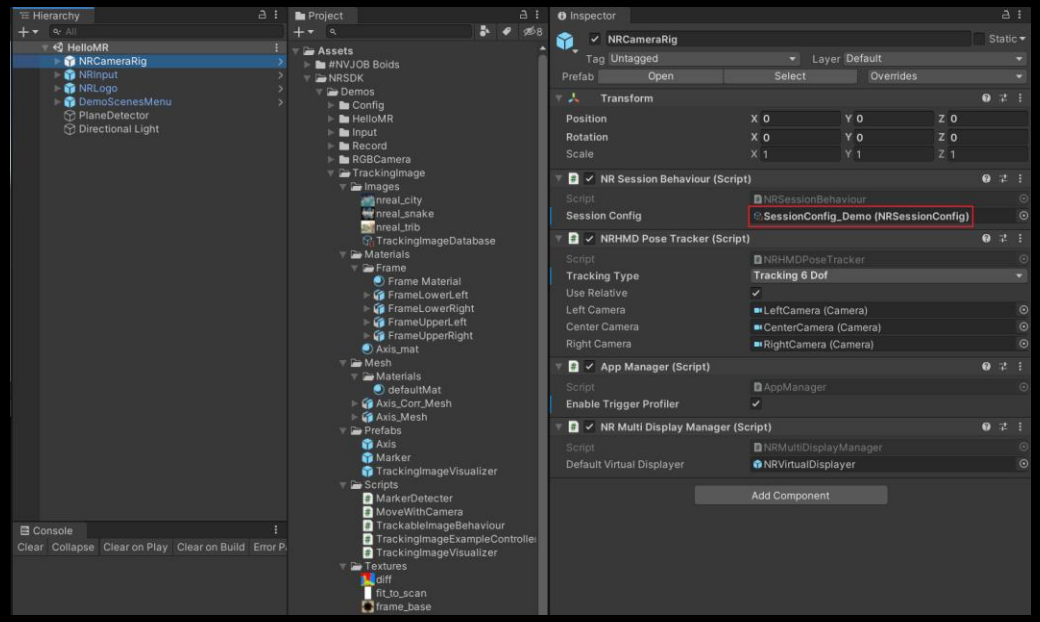
Asset 배치

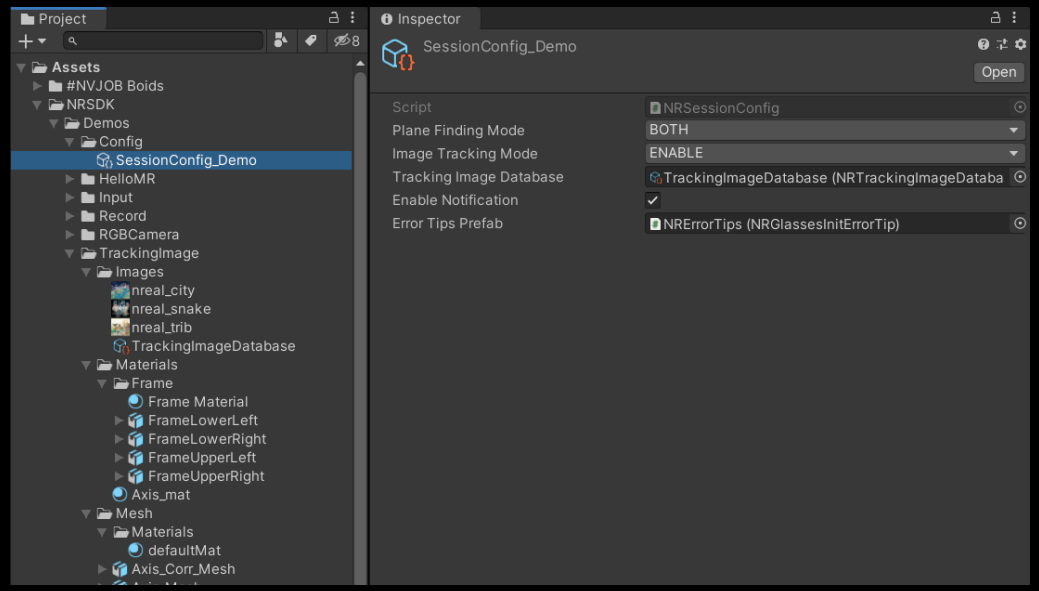

Hierarchy
All
RaisingFish
Directional Light
NRCameraRig
NRInput
Fish 0 Pref
Console
Clear
Collapse
Clear on Play
Clear on B
[09:29:05] [Info][SessionBehaviour
Project
nreal_city
nreal_snake
nreal_trib
TrackingImageDatabase
Materials
Frame
Frame Material
FrameLowerLeft
FrameLowerRight
FrameUpperLeft
FrameUpperRight
Axis_mat
Mesh
Materials
defaultMat
Axis_Corr_Mesh
Axis_Mesh
Prefabs
Axis
Marker
TrackingImageVisualizer
Scripts
MarkerDetecter
MoveWithCamera
TrackableImageBehaviour
TrackingImageExampleControlle
TrackingImageVisualizer
Textures
diff
fit_to_scan
frame_base
HelloMR-3dof
HelloMR-ChangeMode
HelloMR
ImageTracking
Input-ControllerInfo
Input-Interaction
Input-VirtualController
RGBCamera-Capture
Inspector
Fish 0 Pref
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y -0.05 Z 0.5
Rotation X 0 Y 180 Z 0
Scale X 0.1 Y 0.1 Z 0.1
Fish 2 (Mesh Filter)
Mesh Fish 2
Mesh Renderer
Materials
Size 1
Element 0 Fish 0
Lighting
Cast Shadows On
Receive Shadows
Contribute Global Illumination
Receive Global Illumination Light Probes
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusion
Fish 0
Shader #NVJOB/NX Shaders/Boids/Fish
Add Component

2. HelloMR

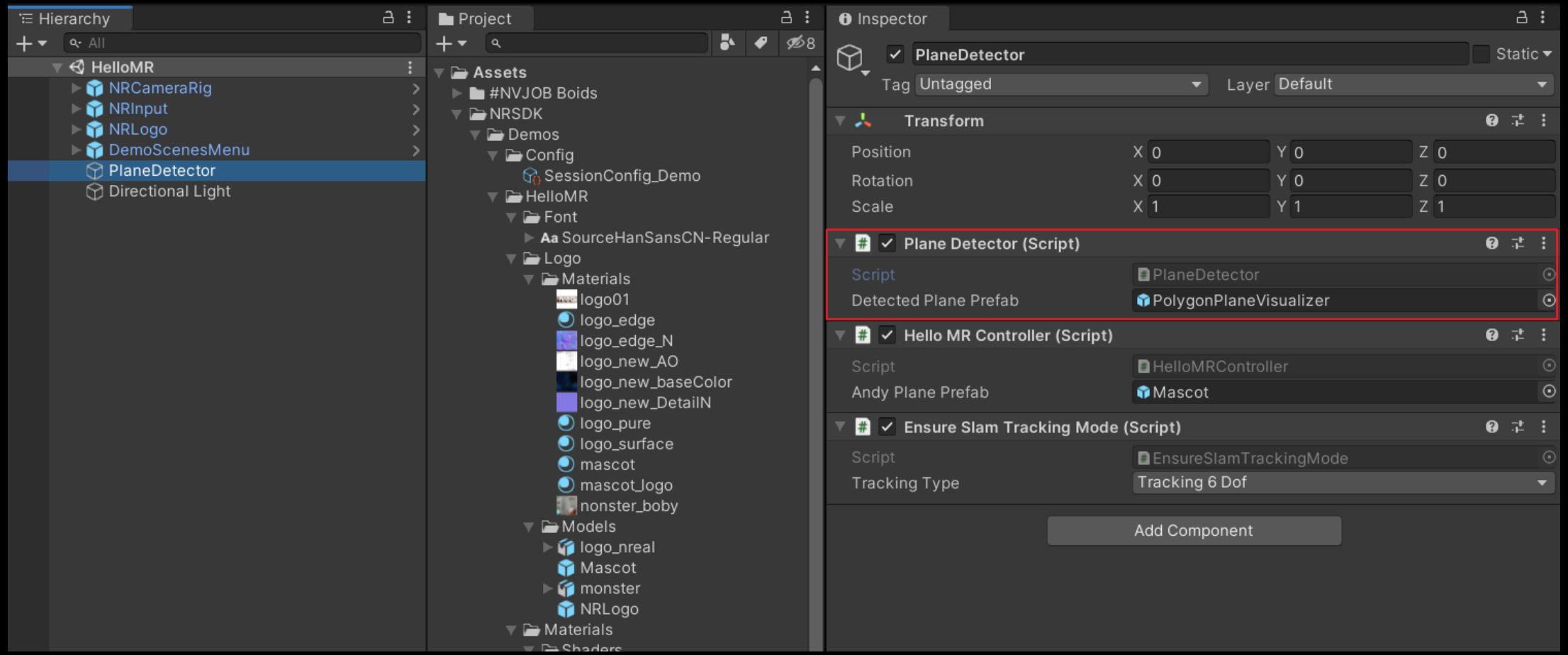

Scene
Shaded
2D
Gizmos
Game
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
Gizmos
Hierarchy
HelloMR
NRCameraRig
NRInput
NRLogo
DemoScenesMenu
PlaneDetector
Directional Light
Console
Clear
Collapse
Clear on Play
Clear on Build
Project
Assets
#NVJOB Boids
NRSDK
Demos
Config
HelloMR
Input
Record
RGBCamera
TrackingImage
Images
nreal_city
nreal_snake
nreal_trib
TrackingImageDatabase
Materials
Frame
Frame Material
FrameLowerLeft
FrameLowerRight
FrameUpperLeft
FrameUpperRight
Axis_mat
Mesh
Materials
defaultMat
Axis_Corr_Mesh
Axis_Mesh
Prefabs
Axis
Marker
TrackingImageVisualizer
Scripts
MarkerDetecter
MoveWithCamera
TrackableImageBehaviour
TrackingImageExampleController
TrackingImageVisualizer
Textures
diff
fit_to_scan
frame_base
HelloMR-3dof
HelloMR-ChangeMode
HelloMR
ImageTracking
Input-ControllerInfo
Input-Interaction
Input-VirtualController
RGBCamera-Capture
RGBCamera-Record
RGBCamera-Yuv
RGBCamera
Editor
Emulator
Materials
Models
Plugins
Prefabs
Resources

NRSessionConfig





Plane Detector

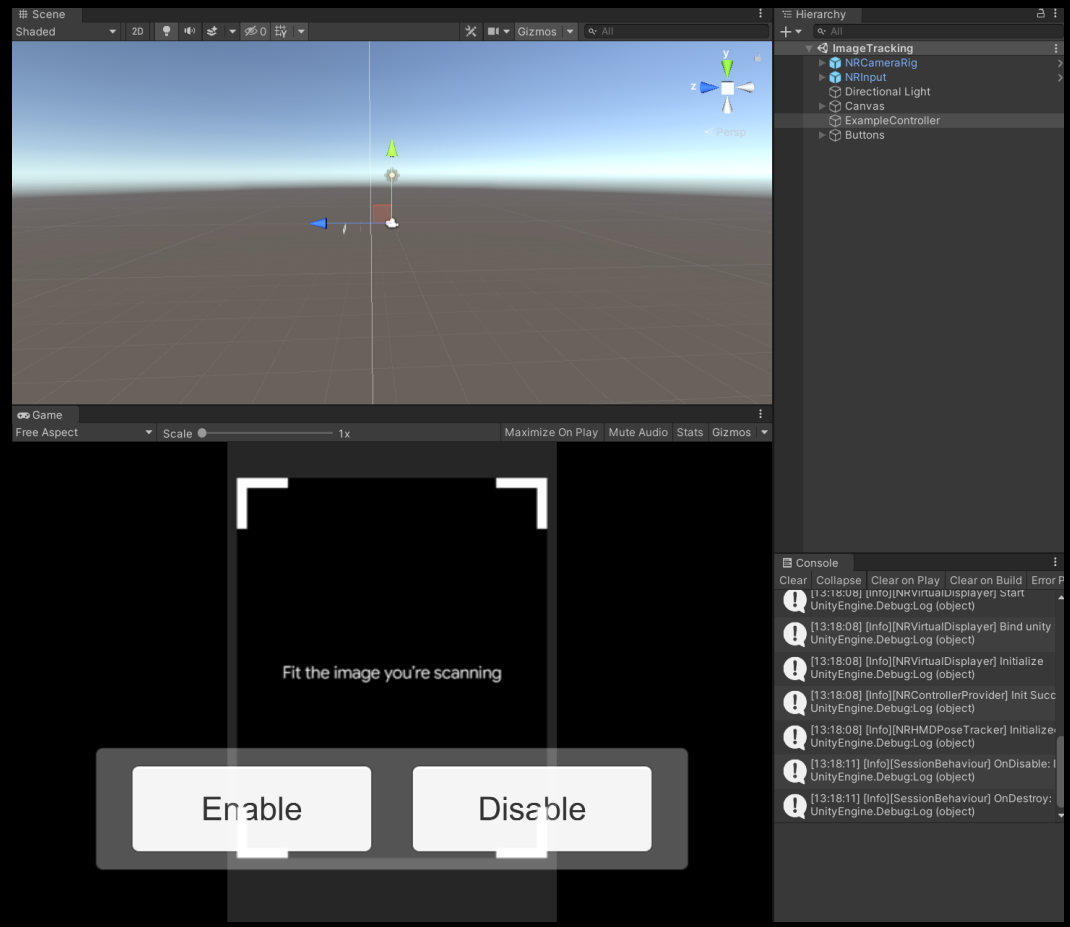


PlaneDetector

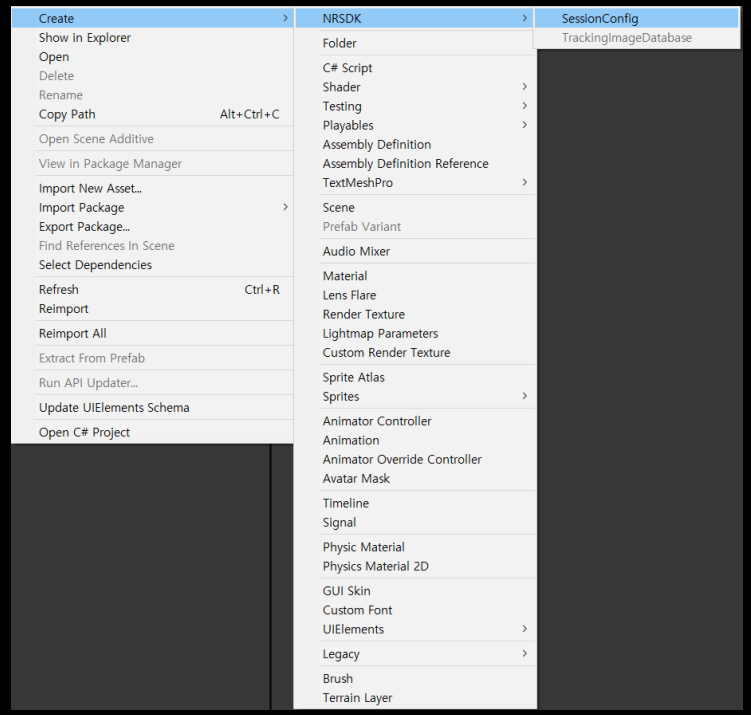
- NRTrackable
- NRInternel이 탐지한 실제 환경에서 추적 가능. 추적 가능한 평면 및 추적 가능한 이미지의 기본 클래스입니다.이 클래스를 통해 응용 프로그램은 추적 가능한 개체의 정보를 얻을 수 있습니다.

- NRTrackablePlane
- NRInternel에 의해 탐지된 실제 세계의 Plane

3. Image Traking

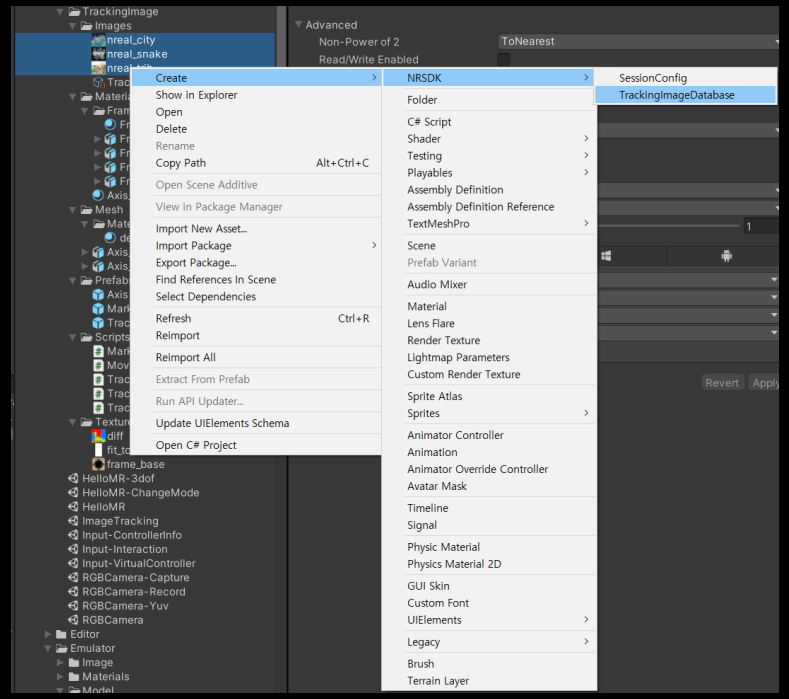


SessionConfig 생성



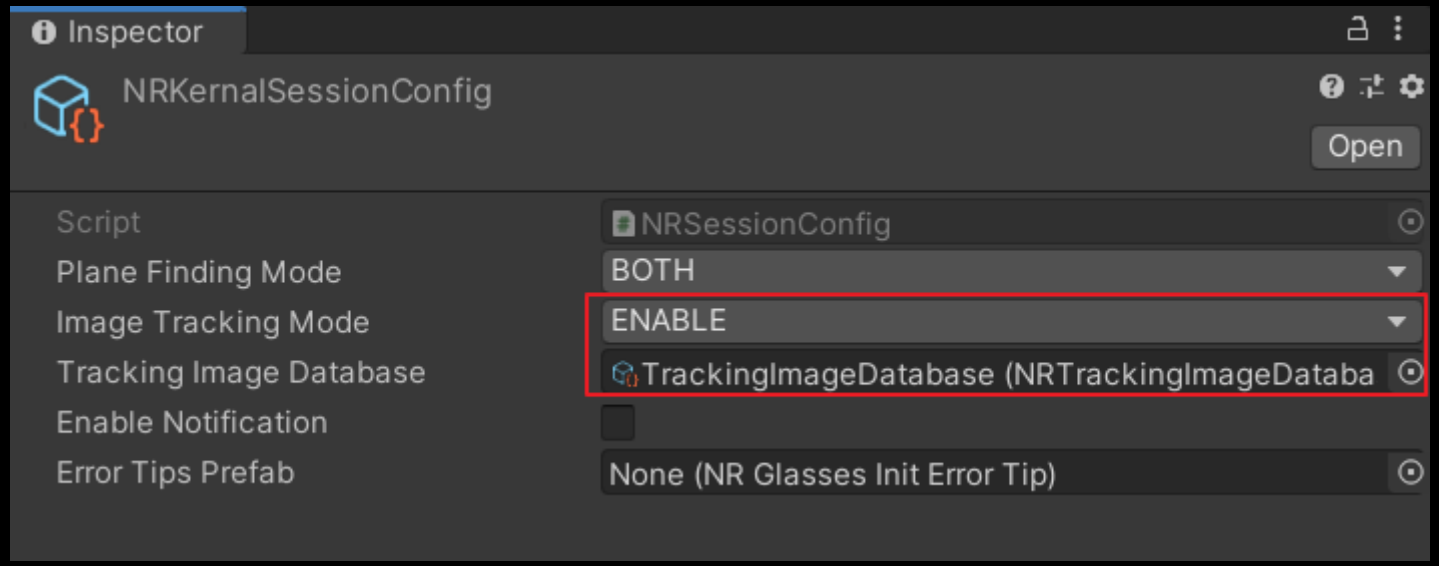
1. Create > NRSDK > SessionConfg를 생성합니다.

TrackingImageDatabase 생성



1. Database를 생성할 이미지를 선택하고, Craete > NRSDK > TrackingIamgeDatabase를 생성합니다.

SessionConfig 설정



1. Image Tracking Mode를 ENABLE로 설정합니다.
2. Tracking Image Database를 설정합니다.

TrackingImageVisualizer

```
/// <summary> Uses 4 frame corner objects to visualize an TrackingImage. </summary>
Unity 스크립트|참조 5개
public class TrackingImageVisualizer : MonoBehaviour
{
    /// <summary> The TrackingImage to visualize. </summary>
    public NRTrackableImage Image;

    /// <summary> A model for the lower left corner of the frame to place when an im ...
    public GameObject FrameLowerLeft;

    /// <summary> A model for the lower right corner of the frame to place when an i ...
    public GameObject FrameLowerRight;

    /// <summary> A model for the upper left corner of the frame to place when an im ...
    public GameObject FrameUpperLeft;

    /// <summary> A model for the upper right corner of the frame to place when an i ...
    public GameObject FrameUpperRight;

    /// <summary> The axis. </summary>
    public GameObject Axis;

    /// <summary> Updates this object. </summary>
    Unity 메시지|참조 0개
    public void Update()
    {
        if (Image == null || Image.GetTrackingState() != TrackingState.Tracking)...

        float halfWidth = Image.ExtentX / 2;
        float halfHeight = Image.ExtentZ / 2;
        FrameLowerLeft.transform.localPosition = (halfWidth * Vector3.left) + (halfHeight * Vector3.back);
        FrameLowerRight.transform.localPosition = (halfWidth * Vector3.right) + (halfHeight * Vector3.back);
        FrameUpperLeft.transform.localPosition = (halfWidth * Vector3.left) + (halfHeight * Vector3.forward);
        FrameUpperRight.transform.localPosition = (halfWidth * Vector3.right) + (halfHeight * Vector3.forward);

        var center = Image.GetCenterPose();
        transform.position = center.position;
        transform.rotation = center.rotation;
```

TrackingImageExampleController

```
@Unity 스크립트 | 참조 0개
public class TrackingImageExampleController : MonoBehaviour
{
    /// <summary> A prefab for visualizing an TrackingImage. </summary>
    public TrackingImageVisualizer TrackingImageVisualizerPrefab;

    /// <summary> The overlay containing the fit to scan user guide. </summary>
    public GameObject FitToScanOverlay;

    /// <summary> The visualizers. </summary>
    private Dictionary<int, TrackingImageVisualizer> m_Visualizers
        = new Dictionary<int, TrackingImageVisualizer>();

    /// <summary> The temporary tracking images. </summary>
    private List<NRTrackableImage> m_TempTrackingImages = new List<NRTrackableImage>();

    /// <summary> Updates this object. </summary>
    @Unity 메시지 | 참조 0개
    public void Update()
    {
#if !UNITY_EDITOR
...
#endif

        // Get updated augmented images for this frame.
        NRFrame.GetTrackables<NRTrackableImage>(m_TempTrackingImages, NRTrackableQueryFilter.New);

        // Create visualizers and anchors for updated augmented images that are tracking and do not previously
        // have a visualizer. Remove visualizers for stopped images.
        foreach (var image in m_TempTrackingImages)
        {
            TrackingImageVisualizer visualizer = null;
            m_Visualizers.TryGetValue(image.GetDataBaseIndex(), out visualizer);
            if (image.GetTrackingState() == TrackingState.Tracking && visualizer == null)
            {
                NRDebugger.Info("Create new TrackingImageVisualizer!");
                // Create an anchor to ensure that NRSDK keeps tracking this augmented image.
                visualizer = (TrackingImageVisualizer)Instantiate(TrackingImageVisualizerPrefab, image.GetCenterPose().position, image.GetCenterPose().rotation);
                visualizer.Image = image;
                visualizer.transform.parent = transform;
                m_Visualizers.Add(image.GetDataBaseIndex(), visualizer);
            }
            else if (image.GetTrackingState() == TrackingState.Stopped && visualizer != null)
            {
                m_Visualizers.Remove(image.GetDataBaseIndex());
                Destroy(visualizer.gameObject);
            }

            FitToScanOverlay.SetActive(false);
        }

    }
```

참고 자료

- #1 How to create an Nreal AR app in Unity in just 8 minutes?
- https://youtu.be/nhpVEsrz5Xc

- #2 How to build and test AR apps right in Unity? | Nreal Emulator
- https://youtu.be/sQApTthKG-k

- #3 How to use image tracking on Nreal Light?
- https://youtu.be/17fNxm1p8-k