



UNITY 3D -CHAPTER1-

SOULSEEK

목차

1.Game Engine

2.Unity 3D

3.Make Project

4.Asset

5.Interface

GAME ENGINE

1. GAME ENGINE

- 게임 개발에 기반이 되는 구성 요소들이 잘 융합된 상태의 소스 코드와 그 기능들을 디자이너들이 사용 가능한 툴을 겸비한 게임 개발 소프트웨어를 일컫는다.
- 둠 시리즈, 퀘이크 시리즈, 언리얼, **F.E.A.R**
 - 이전에는 그 게임만을 위한 작업을 했었고 그것을 다른 게임에 활용할 필요성이 없었다.
 - 라이브러리(**library**) 수준의 프레임 워크(**FrameWork**) 형태였다.
 - 이들 게임이 나오던 시기부터 특정 시리즈, 특정 장르, 대부분의 게임 등등 게임을 만들기 위한 목적으로 **Engine**이라는 모습이 구체화 되었다.
- 현재에 와서는 다양한 기술들이 접목되었고 그래픽 렌더링을 위한 그래픽 엔진과 물리처리를 위한 물리엔진 등등 다양한 기술을 처리하기 위한 기능들을 포함하고 여러 가지 플랫폼까지 지원하는 엔진들이 나오고 있다.

라이브러리 : 흔히 코드 라이브러리라 말하며 사용할 수 있는 코드덩어리(함수나 클래스)를 언제나 사용할 수 있게 준비되어 있는 것을 말한다.
도서관에 당장 보진 않지만 여러 책들이 준비되어 있고 필요한 책이 있을 때 마다 빌려볼 수 있는 것과 같은 것이다.

프레임워크 : 특정 목적에 맞게 만들 수 있도록 미리 준비해둔 뼈대, 틀.

The background features a large, faint, light-blue circular logo, which is the Unity 3D logo, centered on a dark blue gradient. The logo consists of a circle with a stylized 'U' shape inside. In the four corners of the image, there are decorative white line art elements resembling circuit boards or abstract geometric patterns.

UNITY3D

2. UNITY 3D

- 멀티 플랫폼 빌드 : 모바일(**iOS, Android, UWP, Fire OS**), **PC, Console, WebGL, XR(VR/AR/MR)**을 합친 것), 스마트**TV**등 다양한 플랫폼 버전으로 빌드가 가능하다.
- 전 세계 무료화 게임 중 **50%**가 **Unity**로 제작되었다.
- 유니티 에디터에서 디자인, 스크립팅, 디버깅, 빌드 및 테스트 까지 일련의 개발 프로세스가 모두 이루어지는 직관적인 개발 환경(**IDE**)를 제공한다.
- 에셋 스토어를 통해 에셋 패키지 형태의 결과물 공유가 가능해서 다양한 방면의 개발 **Plug-In**들을 사용할 수 있다. 대표적으로 **NGUI**가 있다.
- 일정 수익 이하의 결과물을 내는 콘텐츠는 무료개발이 가능하며 유니티 자체 광고기능을 활용할 수도 있다.

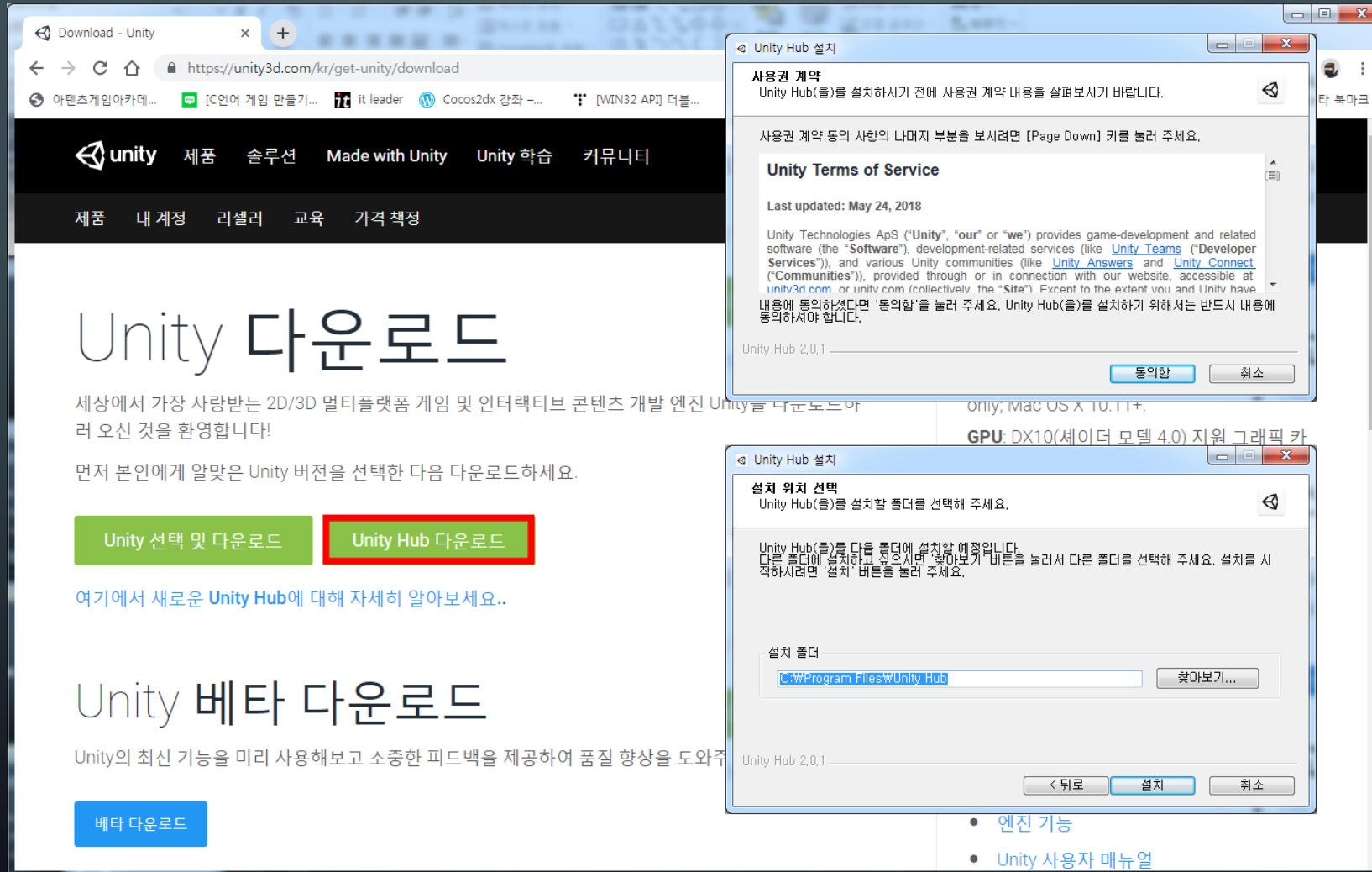
다양한 장점이 있겠지만 궁극적으로 멀티플랫폼으로의 개발 및 빌드, 많은 점유율을 가진 개발 상황, 그 만큼 많은 정보들을 통해 개발의 편리함이 있기 때문에 **Unity**를 선택하게 된다.

MAKE PROJECT

3. MAKE PROJECT

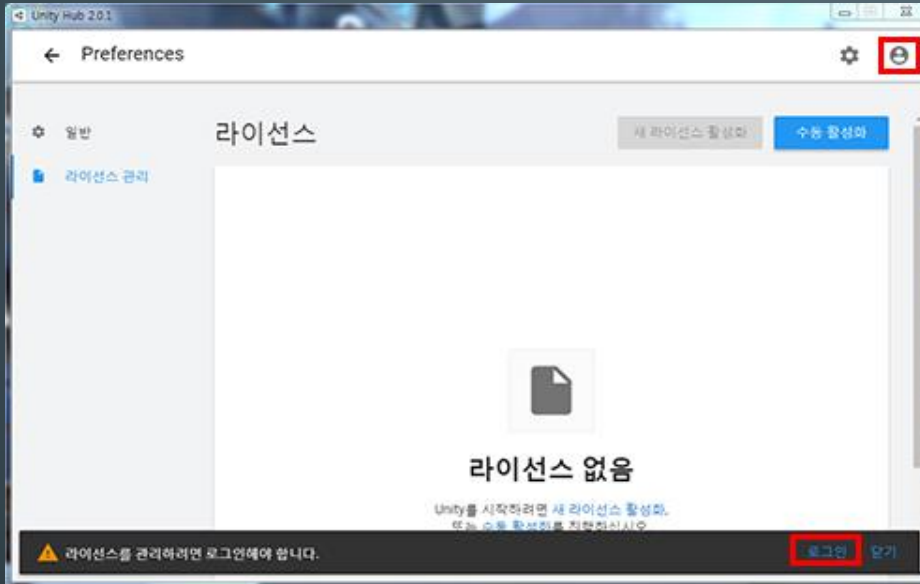
Unity3D 설치하기

- <https://unity3d.com/kr/get-unity/download>로 이동한다.
- **Unity Hub**를 다운로드 받고 나서 설치한다.



3. MAKE PROJECT

Unity3D 설치하기



이미 라이선스가 등록된 ID로 로그인이 되어있으면 최초로 뜨는 창의 형태가 될 것이다!!

1. 설치탭을 눌러주면 **오른쪽 상단의 추가 버튼**을 눌러서 Unity를 설치 할 수 있다.

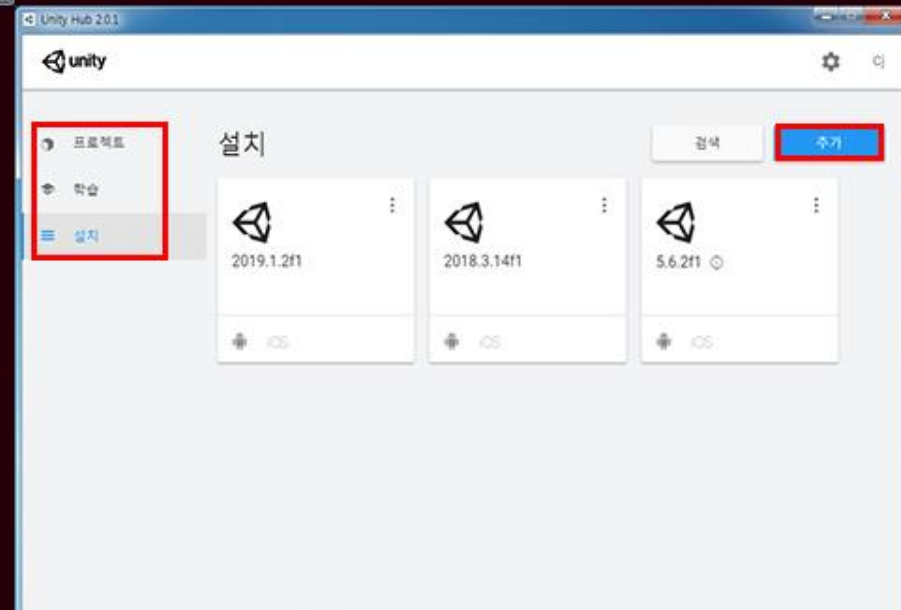
2. 학습탭을 눌러서 Unity에서 준비한 학습 예제를 자료들을 확인할 수 있다.

3. 프로젝트탭을 눌러서 기존의 프로젝트를 불러 올 수 있다.

로그인을 해서 라이선스를 획득하자!!
ID가 없다면 ID를 생성하면 된다.

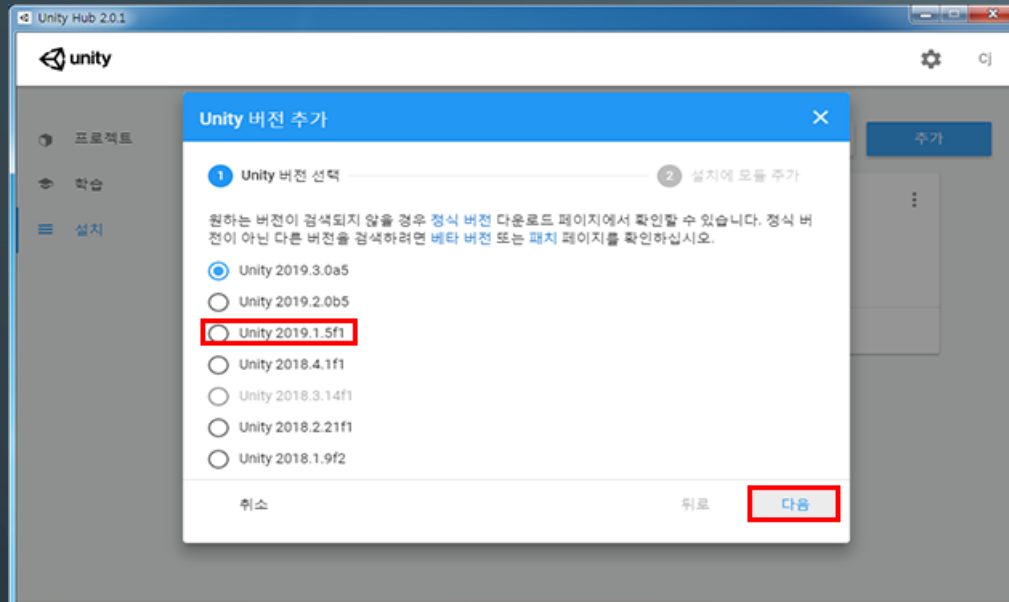
1. **하단의 로그인**글자 혹은 **우측 상단의 아이콘**을 눌러서 로그인이 가능하다.

2. 로그인을 완료한 후 **뒤로가기 버튼**을 눌러준다.



3. MAKE PROJECT

Unity3D 프로젝트 만들기



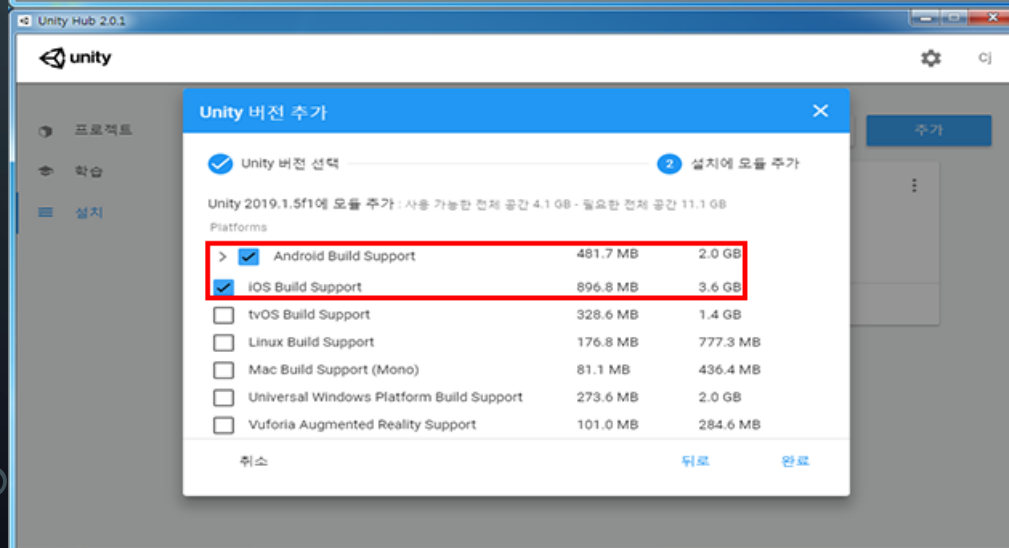
1. 알맞는 설치버전을 선택하자.

버전에 a, b가 들어간것은 알파, 베타버전이라는 뜻이다.
프로젝트 버전과 달라도 되는 부분은 첫자리와 중간자리는 같은 버전이라면 큰 오류 없이 실행이 가능하지만 딱 맞추는 것이 가장 좋다.

2. 플랫폼 선택 화면에서 아무 선택없이하면 데스크탑에서 실행되는 exe파일만 빌드가능하다.

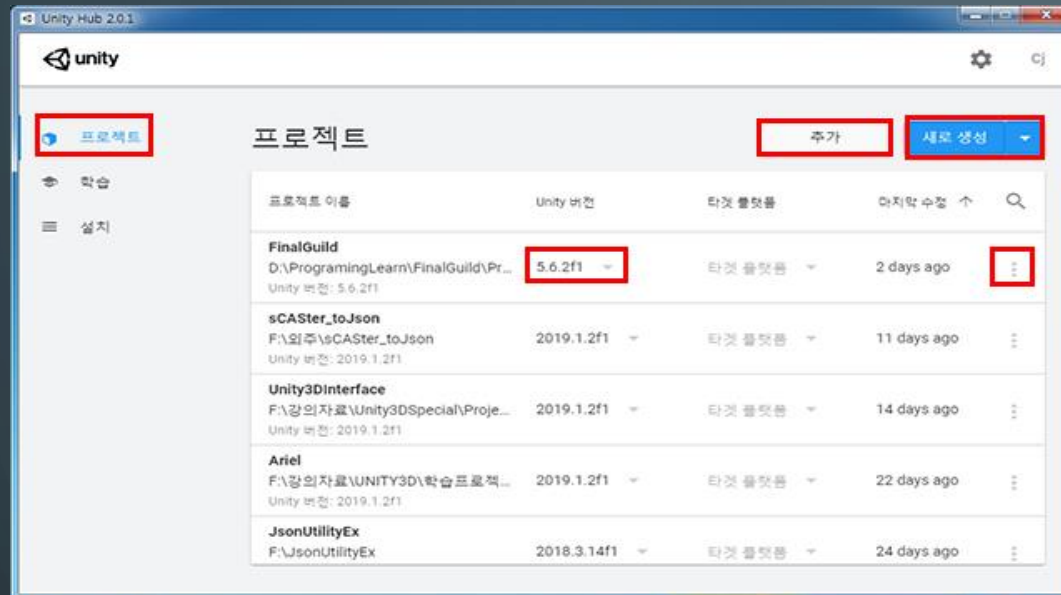
현재 모바일은 Android, iOS 두가지의 운영체제를 가지고 있으므로 **Android, iOS를 설치**해두자!!

3. Documentation은 기본 선택사항이고 한국어 언어팩을 추가해준뒤 완료 버튼을 눌러서 설치를 진행하자.
기본 빌드 도구인 Visual Studio 2017 Community가 설치되어 있지 않다면 함께 설치하기때문에 설치시간이 길수도 있다.



3. MAKE PROJECT

Unity3D 프로젝트 만들기



1. Unity Hub를 통해서 작업을 진행한 프로젝트들은 리스트로 나올 것이고 그렇지 않은 것들은 “추가” 버튼을 통해서 추가 할 수 있다.

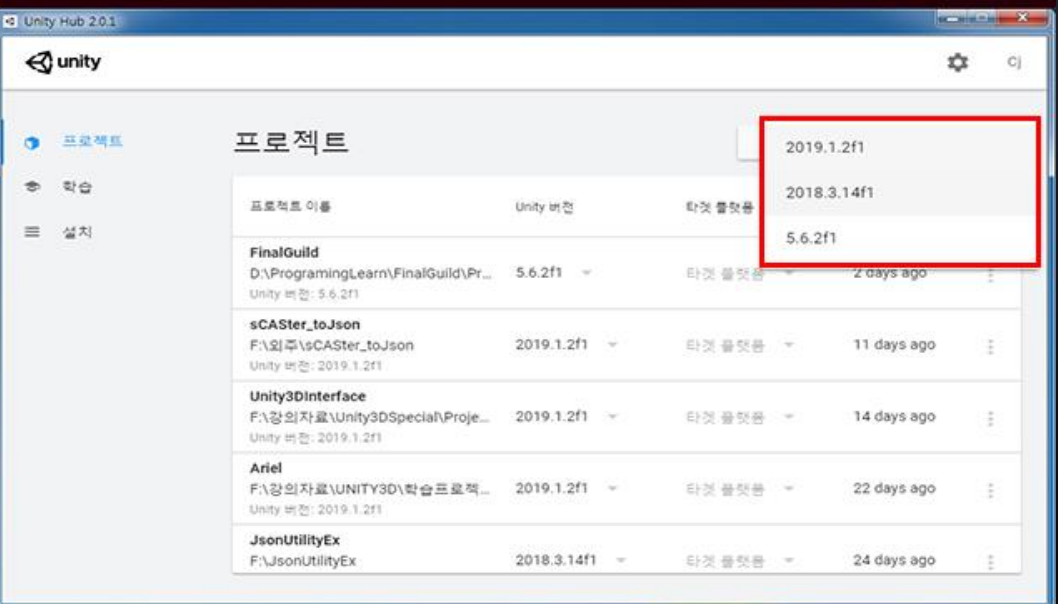
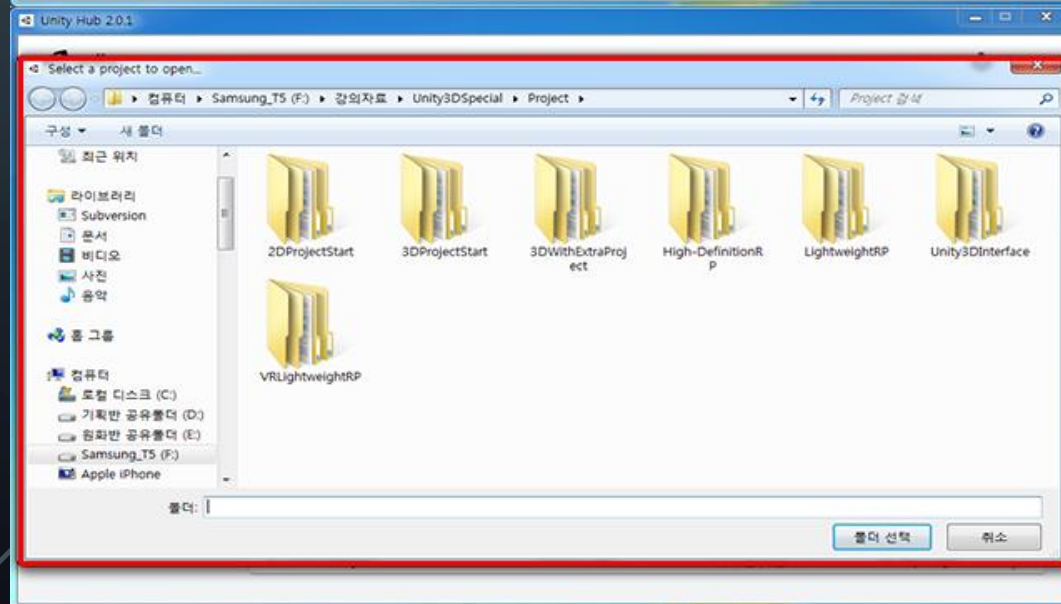
2. 프로젝트 리스트에서 버전표시에 해당 버전 Unity로 실행 할 수 있게 변경 할 수 있다.

3. 프로젝트 리스트 오른쪽 끝에 있는 버튼으로 프로젝트 폴더로 가는

4. 프로젝트 폴더를 선택할 때에 하위폴더에 Asset폴더가 있는 곳에서 추가를 실행해줘야한다.

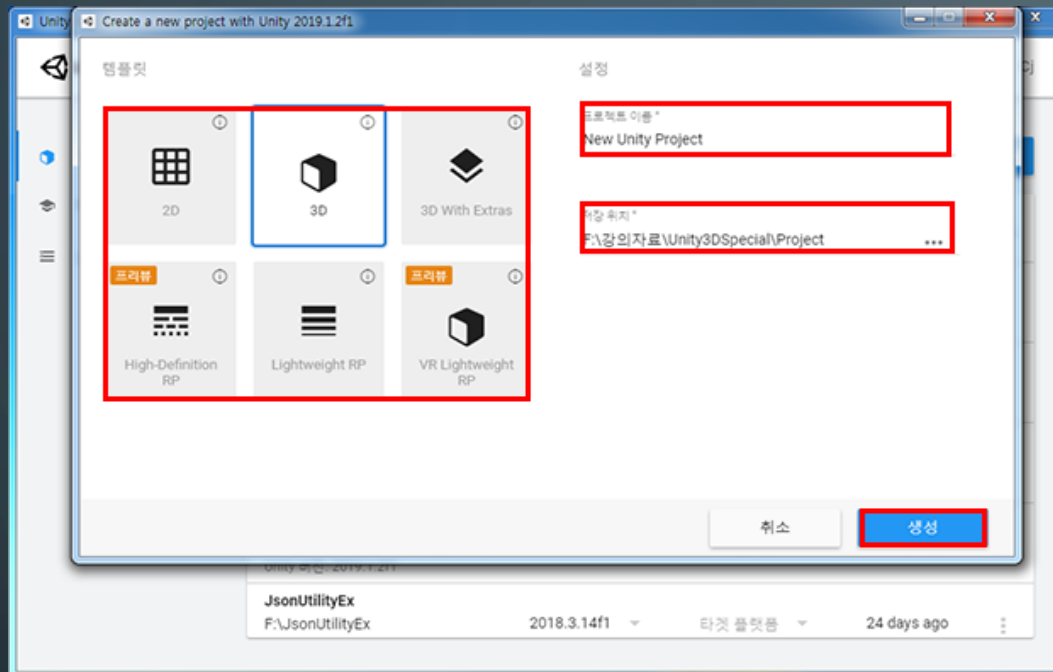
5. “새로생성” 버튼을 선택하면 프로젝트를 새로 생성하게 된다.

6. “새로생성” 버튼 옆의 화살표 버튼으로 자신에게 설치되어 있는 Unity버전이 다양하다면 타겟 버전을 선택할 수 있다.



3. MAKE PROJECT

Unity3D 프로젝트 만들기



2. 프로젝트명을 입력한 뒤,
저장위치 오른쪽에 있는 아이콘을 눌러서 경로를 지정해주면
해당프로젝트는 지정경로에 생성된다.
생성 버튼을 누르면 설정한 프로젝트에 맞게 생성이 된다.

1. 초기 셋팅을 어떤 작업을 위주로 할 것인지에 따른 프로젝트 초기 셋팅을 설정한다. 초기 선택은 그냥 선택일 뿐이다 어차피 엔진에서 모든 기능을 사용할 수 있다.

2D : 2D 프로젝트를 만들기 위한 기본 셋팅 된 상태의 프로젝트를 생성해준다. **View와 Camera가 2D작업에 맞는 셋팅**이다. 이 상태에서 3D로 다시 돌려 놓을 수 있다.

3D : 3D 프로젝트를 만들기 위한 기본 셋팅 된 상태의 프로젝트를 생성해준다. **View와 Camera가 3D 작업에 맞는 셋팅**이다. 이 상태에서 2D로 전환이 가능하다.

3D With Extras : **Post Processing Stack이 추가된 상태로** 프로젝트를 생성해주며, 이기능이 어떤것인지 볼 수 있는 예제 Scene과 함께 생성된다.

High-Definition RP : **고화질 스크립팅이 가능한 파이프 라인**을 사용. Shader Model 5.0(DX11 이상)을 지원하는 플랫폼용 게임을 개발하려는 고급 그래픽에 중점을 둔 프로젝트를 지원한다. **Shadergraph, Post Processing V2 Statck** 기능이 적용되어 있고, 기술을 보여주는 예제 Scene과 함께 생성된다.

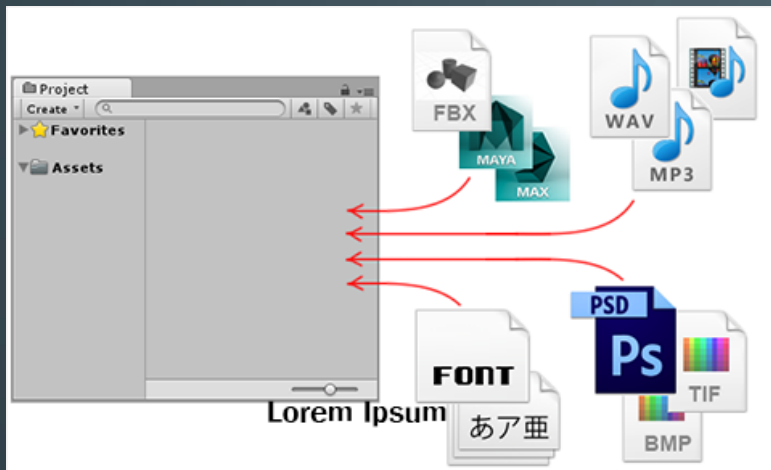
Lightweight RP : **저화질 혹은 중간등급의 스크립팅이 가능한 파이프 라인**을 사용. Shadergraph, Post Processing V2 Statck 기능이 적용되어 있고, 기술을 보여주는 예제 Scene과 함께 생성된다.

VR Lightweight RP : Lightweight RP 프로젝트를 **VR 환경에 맞게 셋팅**을 변경한 프로젝트를 생성한다.



ASSET

4. ASSET



Asset은 게임이나 프로젝트에서 사용될 수 있는 **다양한 아이템**을 말한다. Asset은 3D모델이나 오디오 파일, 이미지 또는 Unity가 지원하는 다른 형식의 파일등 **Unity외부에서 생성된 파일에서 가져올수 있다.** 또한 애니메이터 컨트롤러, 오디오 믹서 및 렌더 텍스처와 같이 **Unity에서 생성할 수 있는 일부 타입의 에셋**도 있다.

일반적인 에셋 타입

이미지 파일 : BMP, TIF, TGA, JPG, PSD등 가장 일반적인 이미지 파일 타입 대부분을 지원한다.

FBX 및 모델링 파일 : FBX 파일 포맷을 지원하므로 FBX를 지원하는 3D 모델링 소프트웨어에서 데이터를 임포트 할 수 있다.

메시와 애니메이션 : 사용하는 3D 소프트웨어 모델과 관계없이 Unity는 메시와 애니메이션을 각 파일에서 임포트 한다.

오디오 파일 : 압축되지 않은 오디오 파일을 Asset폴더에 저장하면 지정된 압축 설정에 따라 파일이 Unity로 임포트된다.

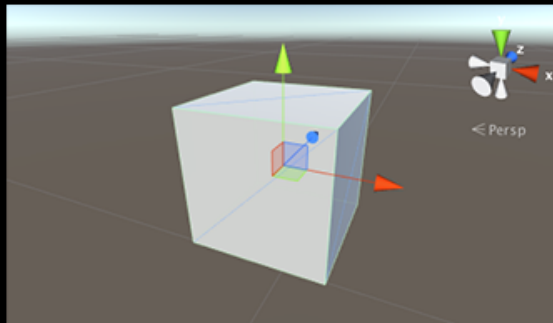
스탠다드 Asset : Unity에서 기본적으로 제공되고 있는 Asset들을 말한다.

4. ASSET

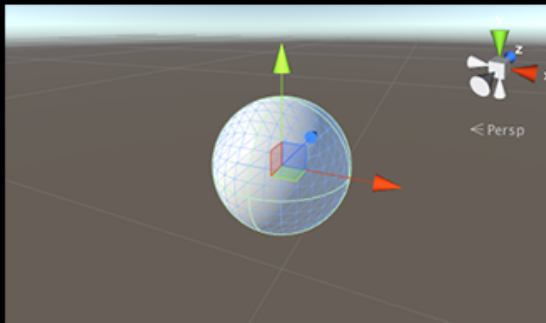
Primitive & Placeholder Object

모델링 소프트웨어로 만들 수 있는 형태의 3D모델을 사용하여 작업하요 임폴트할 수 있지만 Unity에서는 **Cube, Sphere, Capsule, Cylinder, Plane, Quad**등의 **Primitive 오브젝트** 타입이 준비되어 있으며 차후 완성된 3D모델링을 적용하기 전까지 임시로 사용할 3D오브젝트로 사용할 수 있다. 상단 메뉴에서 **GameObject -> 3DObject** 메뉴에서 해당 **Primitive Object**를 **Scene**에 추가 할 수 있다.

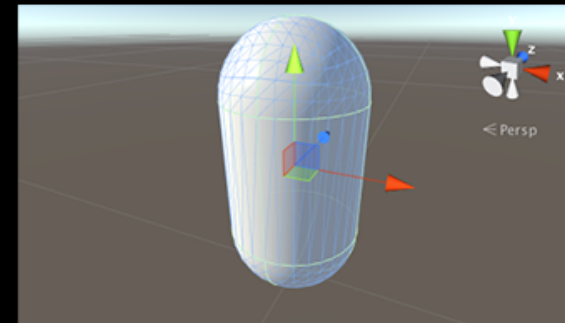
큐브(Cube)



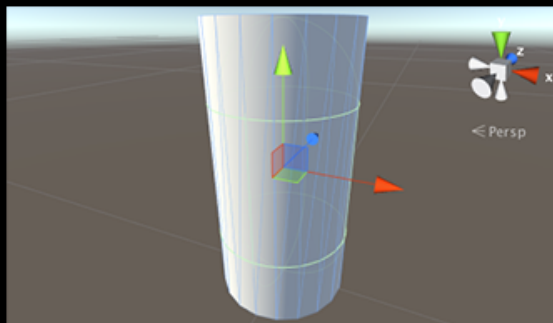
Sphere(구체)



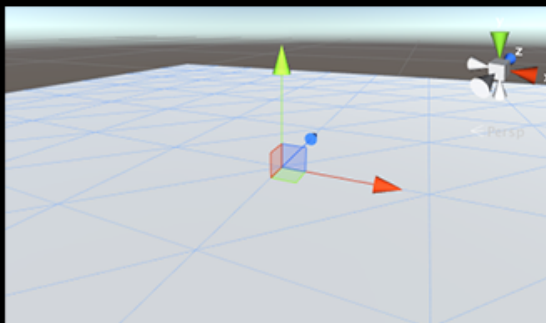
Capsule(캡슐)



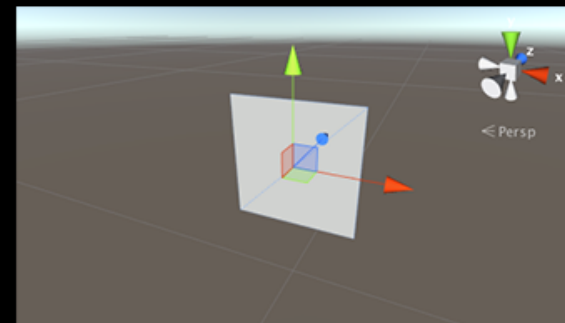
Cylinder(실린더)



Plane(평면)



Quad(사각형)



4. ASSET

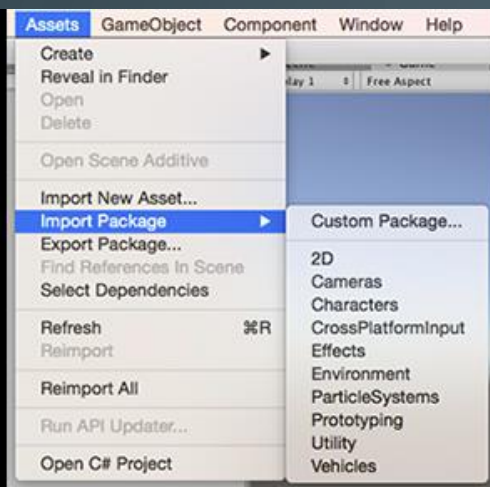
Package

- **Asset Package**와 **Unity Package**가 있다.
- **Asset Package** : **Unity Asset Store**에서 제공되고, **Unity Project**와 **Asset Collection**을 공유하고 재사용하는 기능을 지원한다.

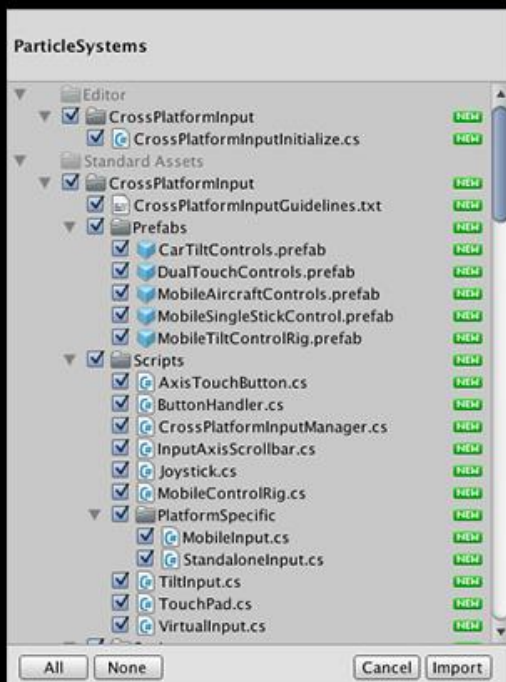
Asset Package

- **Unity Standard Asset**과 **Unity Asset Store**의 아이템 패키지로 제공된다.
- 패키지는 **Unity Project**의 파일 및 데이터 컬렉션 또는 프로젝트 요소로, **zip** 파일과 유사하게 파일 하나로 압축되어 저장된다.
- 패키지의 압축을 풀면 **zip**파일과 마찬가지로 원래의 디렉토리 구조와 **Asset**에 대한 메타 데이터(임포트 설정과 다른 **Asset**으로 연결되는 링크 등)가 유지된다.
- **Export Package** 명령을 통해 압축 및 저장을 하여 패키지를 만들어낸다.
- **Import Package** 명령을 통해 현재 열려 있는 **Unity Project**에 컬렉션의 압축을 푼다.

4. ASSET



Standard Asset Package 와 Custom Package를 Import할 수 있다.
Asset -> Import Package를 선택하면 설정할 수 있다.



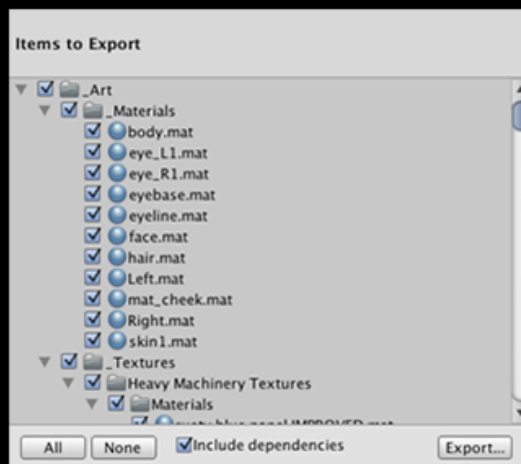
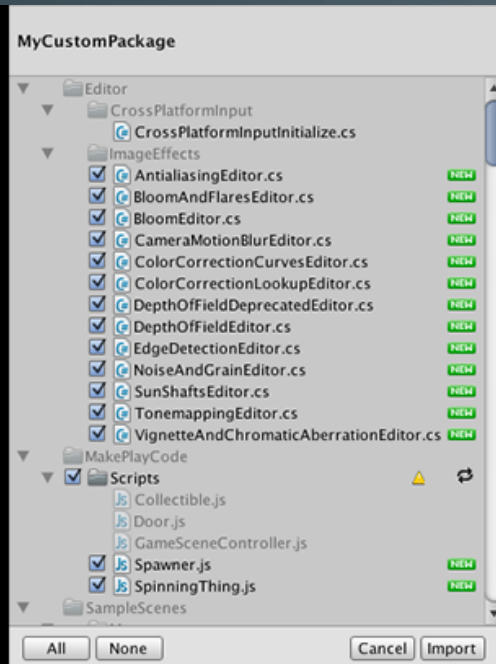
Standard Asset Package

2D, Cameras, CrossPlatformInput, Effects, Prototyping, Utility, Vehicles가
Unity의 **StandardAsset 패키지**이다.

Standard Asset Import

1. 에셋을 임포트할 프로젝트를 열거나 활성화
2. **Asset->Import Package**를 선택한 다음 임포트할 패키지의 이름을 리스트에서 선택
3. Import를 선택하면 Unity는 패키지의 콘텐츠를 **스탠다드 에셋 폴더에 저장하며, Project View에 액세스**할 수 있다.

4. ASSET



Custom Asset Package Import

직접 만든 프로젝트나 다른 Unity 사용자가 만든 프로젝트에서 Export된 Custom Asset Package를 Import할 수 있다.

1. Asset을 Import할 프로젝트를 연다.
2. Asset -> Import Package -> Custom Package...를 선택
3. 탐색기에서 원하는 패키지를 선택하면 Import Unity Package Dialog가 패키지의 모든 항목이 미리 선택되어 설치할 준비가 된 상태로 표시된다.
4. Import를 선택하면 Unity는 패키지의 콘텐츠를 Assets폴더에 저장하며, Project View에서 액세스 할 수 있다.

Package Export

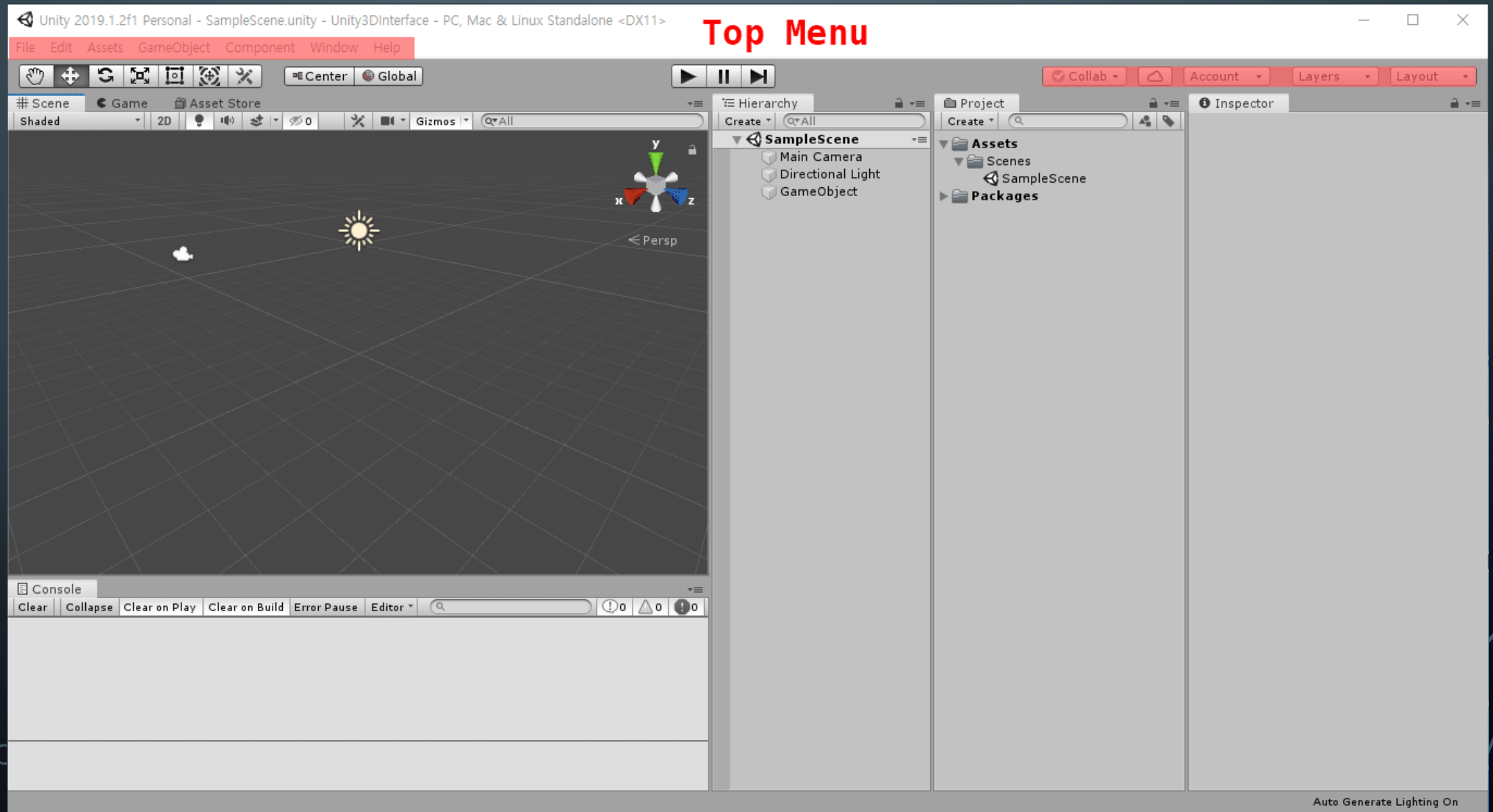
Export Package를 사용하여 프로젝트 일부 혹은 전체를 CustomPackage로 만들어야 한다.

1. Asset을 Export할 프로젝트를 연다.
2. 메뉴에서 Asset->Export Package...를 선택하여 Exporting Package Dialog를 연다.
3. 다이얼로그에서 패키지에 포함시킬 Asset에 해당하는 확인란을 클릭하여 Asset을 선택
4. 선택한 Asset에서 사용하는 모든 Asset이 자동으로 선택되도록 include dependencies 박스를 선택된 상태로 유지한다.
5. Export를 클릭하여 파일탐색기를 실행한 후 패키지 파일을 저장할 곳을 선택
6. 패키지 이름을 지정하고 원하는 위치에 패키지를 저장

INTERFACE

4. INTERFACE

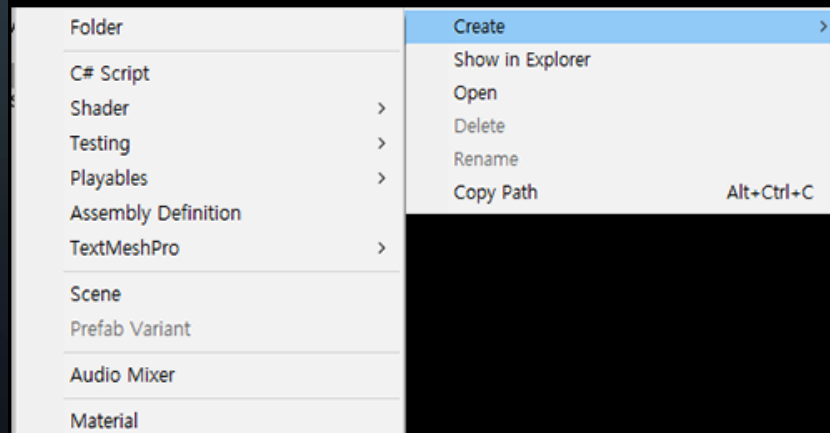
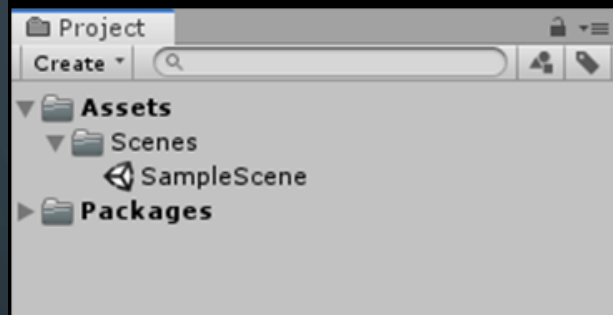
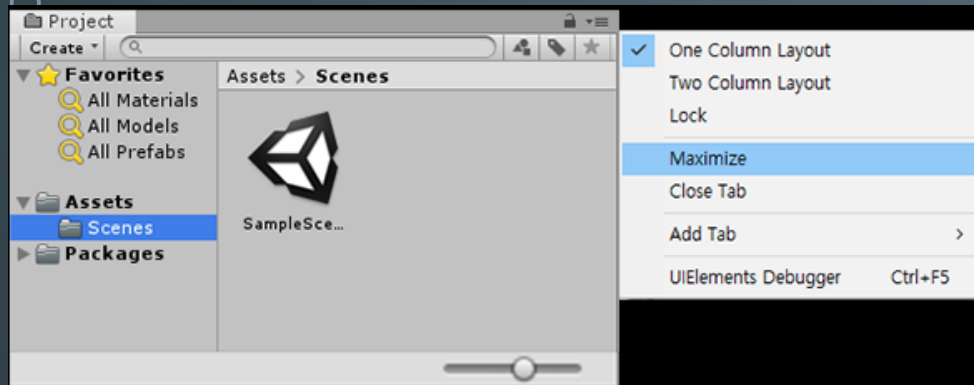
Unity Interface



4. INTERFACE

- **Scene View : Scene**을 시각적으로 탐색하고 편집할 수 있다. **Scene View**에는 작업중인 프로젝트 타입에 따라 **3D or 2D** 원근이 표시된다.
- **Project View : Project**에서 사용할 수 있는 에셋 라이브러리가 표시된다. 프로젝트로 에셋을 임포트하면 이 창에 표시된다.
- **Hierarchy View : Scene**의 모든 오브젝트를 계층 구조에 텍스트로 표시한다. **Scene**에 있는 각 항목마다 **Hierarchy**에 엔트리가 있으므로 두 창은 본질적으로 연결되어 있다고 볼 수 있다. 이 구조는 오브젝트의 상호 연결 방식(부모 - 자식 관계)을 구조로 나타낸다.
- **Inspector View** : 현재 선택된 오브젝트의 모든 프로퍼티를 보고 편집 할 수 있다. 여러 오브젝트 타입마다 서로 다른 여러 프로퍼티가 있으므로 인스펙터 창의 레이아웃과 콘텐츠는 다양하다.
- **Tool bar** : 가장 필수적인 작업기능에 접근할 수 있게 해준다. 일부 **Scene View**에 대한 버튼도 존재한다.
- **Game View** : 상단의 실행버튼을 활성화 하면 **Game View**에 카메라로 현재 **Randering**되어지고 있는 **Scene**를 보여준다.
- **Asset Store** : 유저들에 의해 제작된 **Custom Asset**을 구매할 수 있다. 무료도 꽤 많다.

4. INTERFACE



Project View

Project에 속한 Asset에 액세스하고 Asset을 관리할 수 있다.

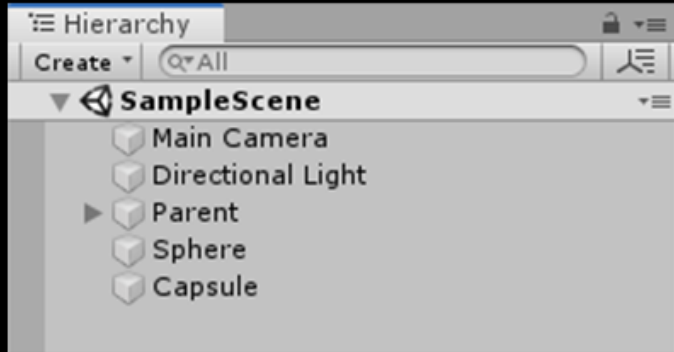
오른쪽 상단에 있는 아이콘을 통해서 **표현 방식을 바꿀 수 있다.**

Two Column 일때, 오른쪽 하단 슬라이드 바로 **아이콘의 크기조절을 가능하게 한다.**

Two Column 일때, 프로젝트 Asset 목록 위에 자주 사용하는 항목을 저장하여 쉽게 액세스 할 수 있는 Favorites 섹션이 있다.

Asset을 관리하는 창이기 때문에 패널위에서 우클릭을 하면 Asset을 생성할 수 있다.

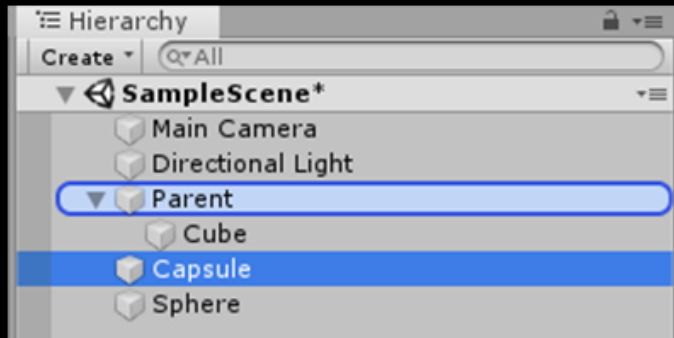
4. INTERFACE



Unity에서는 **부모 - 자식** 이라는 개념으로 Hierarchy가 이루어 진다. 오브젝트 최상단 오브젝트나 Scene이 “**부모**” 가 되고 아래에 있는 모든 오브젝트는 “**자식**” 이 된다. 자식 오브젝트에서 다시 오브젝트를 생성하여 부모 - 자식 관계가 중첩되도록 할 수 있다.

부모와 자식은 Scene상에서 Position, Rotate, Scale의 속성에 그대로 영향을 받고 기준점이 부모가 된다.

이미 생성된 오브젝트끼리도 부모 - 자식 관계를 만들어 줄 수 있다.

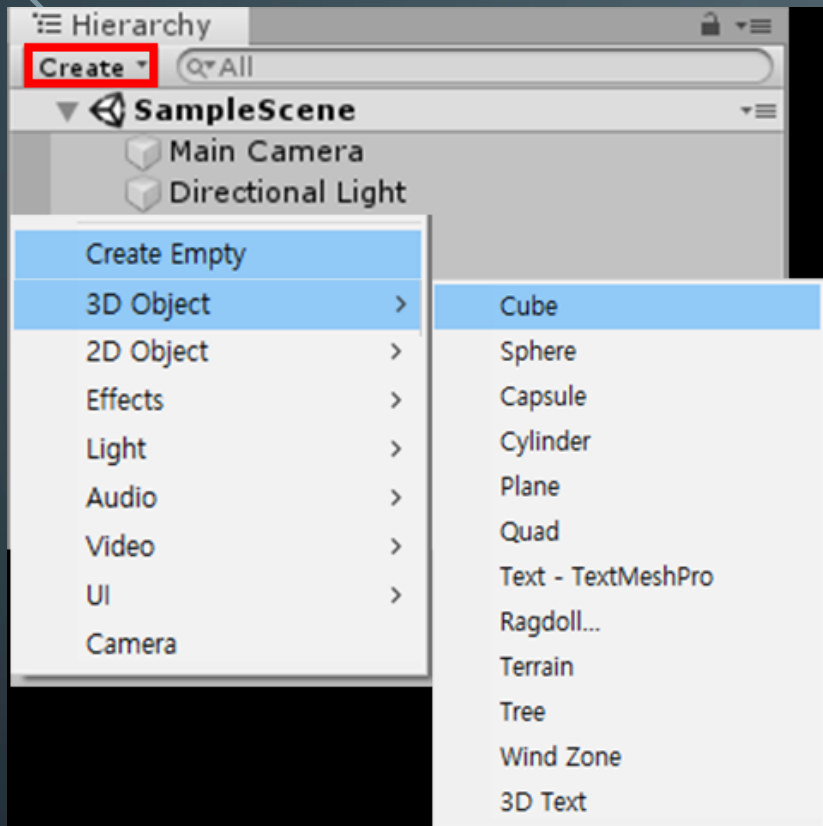


자식이 될 오브젝트를 부모가 될 오브젝트에 드래그해서 올려놓으면 **사각형 파란 박스**가 생기고 그렇게 부모와 자식의 관계를 만들 수 있다



자식이 될 오브젝트를 부모가 될 오브젝트 하위리스트에 드래그해서 포함시킬 수 있으며 **선으로 표시**되면 해당 하위 리스트의 아래순서(자식이 아닌)로 포함된다

4. INTERFACE



Hierarchy View

현재 **Scene에 포함된** 모든 GameObject의 리스트를 보여준다.

오브젝트들은 3D모델과같이 에셋 파일의 **직접적인 인스턴스**이거나 게임 대부분을 차지하는 CustomObject인 **Prefab의 인스턴스**이다.

Scene에 Object가 추가되거나 **제거**되면 Hierarchy에서도 똑같이 해당 Object가 표시되거나 사라진다.

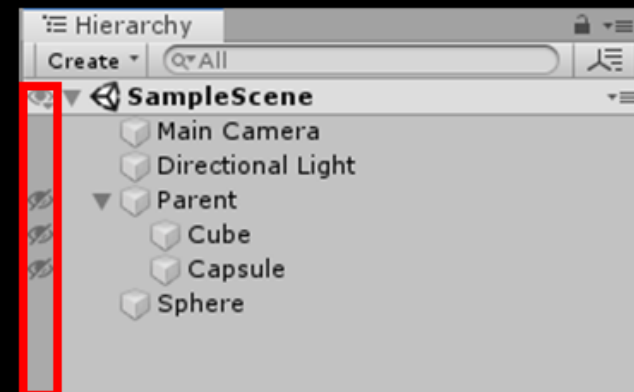
View에서 **우클릭**을 하면 GameObject들을 추가할 수 있다.

View에서 **왼쪽 상단**의 Create 버튼을 클릭해서 GameObject를 추가 할 수 있다.

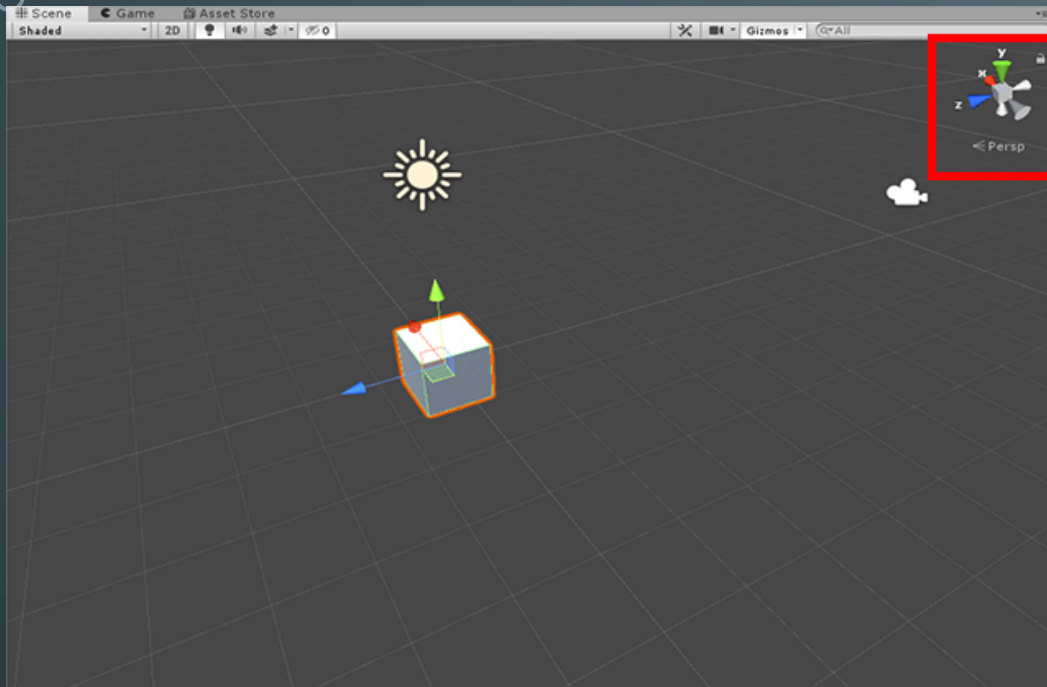
Hierarchy View 왼쪽라인에 활성화와 비활성화를 할 수 있도록 공간이 만들어져 있다.

Photoshop에서 레이어에 있는 눈모양의 아이콘과 같은 기능을 하는 것이다.

비활성화 상태에서는 코드단위, 혹은 속성창에서의 활성화 변경으로 이상태를 벗어날수 없다.



4. INTERFACE



Scene View

작성중인 세계에 대한 **interactive view** 이다.

Scene View를 사용하여 경치, 문자, 카메라, 조명 및 기타 모든 유형의 **게임 객체**를 **선택**하고 **배치** 할 수 있다.

Scene View에서 오브젝트를 선택, 조작 및 수정할 수 있다는 것은 Unity에서 작업을 시작하기 위해 배워야하는 몇 가지 기술이다.

Scene Gizmo

Scene View의 오른쪽 위 모서리에 있다, 카메라의 현재 방향이 표시되고 보기 각도 및 투영 모드를 빠르게 수정 할 수 있다. 큐브의 양쪽에 원주형 팔이 있고, 각 X, Y, Z레비블이 표시되어 있다. 각 원주형 팔을 클릭하면 각 좌표가 나타내는 뷰형태로 카메라가 전환된다 (3D Max의 기능과 비슷하다), 오른쪽 클릭을하면 어떤 방향의 뷰를 볼것인지 설정 할 수 있다.

Gizmo아래 Persp, iso를 각각 클릭으로 전환하면 Perspective를 켜거나 끌 수 있다. 각 원근감을 주는 것과 안주는 것이 되는 것이다. 각 좌표의 원뿔형 팔을 클릭해도 원근감없는 각 좌표방향의 뷰가 된다.

Gizmo 오른쪽 상단에 자물쇠 아이콘은 Scene에서 카메라 회전을 허용할것인지 말것인지를결정하는 것이다.

4. INTERFACE

Hand Tool

핸드툴을 선택하면(단축키: Q) Scene에서 마우스 카메라 컨트롤이 가능하다.



Move: 클릭하고 드래그하여 카메라를 이리저리 끌어 옮긴다.



Orbit: Alt를 누른 상태에서 왼쪽 클릭하고 드래그하면 카메라가 현재 피벗 포인트 주위의 궤도를 따라 회전합니다.



Zoom : Alt 키를 누른 상태에서 마우스 오른쪽 버튼을 클릭하고 드래그 하여 Scene View를 확대 축소 한다.

Flythrough mode

Scene View 마치 1인게임 하듯 비행하며 탐색하는 모드

오른쪽 마우스 버튼을 클릭한 상태로 계속 누르며 움직인다.

마우스를 사용하여 뷰를 이리저리 움직이고 WSAD키를 사용하여 좌우와 앞뒤로 이동하고 Q 및 E를 사용하여 상하로 이동한다.

Shift를 계속 누르고 있으면 더 빠르게 이동한다.

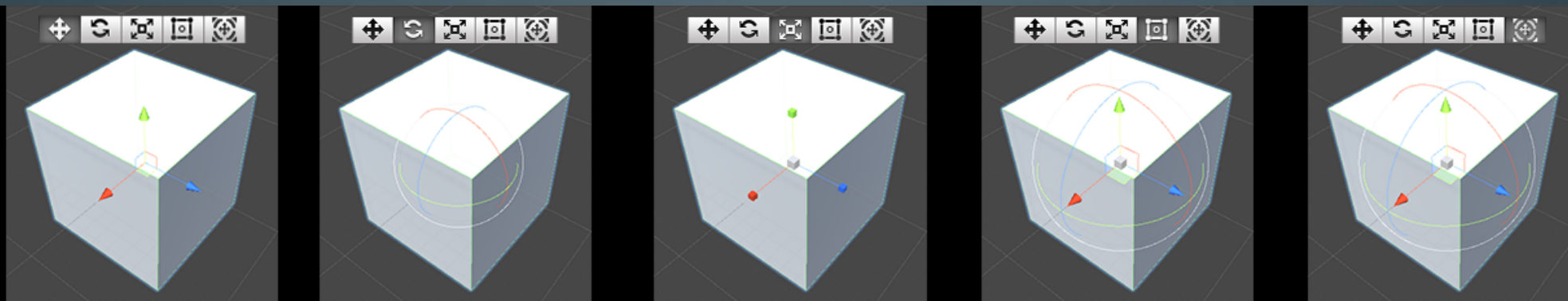
단축키

Move : Alt + 마우스 가운데 버튼 + 드래그

Orbit : Alt + 마우스 왼쪽 버튼 + 드래그

Zoom : 스크롤 휠을 사용하거나, Alt키와 마우스 오른쪽 버튼을 누른 상태로 드래그

4. INTERFACE



Move, Rotate, Scale..

첫 번째 부터 Move, Rotate, Scale, Rect Transform, Transform 도구를 사용하여 개별 게임 오브젝트를 편집할 수 있다.

이동(Move): Move Gizmo 가운데에는 단일 평면 안에서 GameObject를 드래그하는 데 사용할 수 있는 작은 삼각형이 세 개 있다. 한 축이 가만히 정지된 두 축을 한꺼번에 이동할 수 있다. Shift키를 누른채로 Gizmo의 가운데를 클릭하고 드래그하면 가운데 납작한 사각형이 변한다. **단축키는 W**

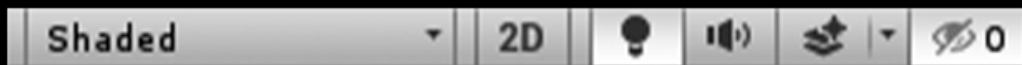
외전(Rotate): 구체 Gizmo축을 선택하고 드래그하여 오브젝트의 외전을 변경한다. **단축키는 E**

스케일(Scale): Scale 도구를 사용하여 Gizmo 가운데에서 큐브를 클릭하고 드래그하여 게임 오브젝트를 모든 축에서 균일하게 다시 Scale 할 수 있다. **단축키는 R**

사각트랜스폼(RectTransform): 2D요소의 위치를 지정할때 많이 사용한다. **단축키는 T**

트랜스폼(Transform): Move, Rotate, Scale 도구가 합쳐진 도구이다. **단축키는 Y**

4. INTERFACE



DrawMode(Render Mode)

Scene View에서 오브젝트들이 렌더링 되어 보여진 모드를 선택.

Shaded : 텍스처가 보이는 표면을 표시

Wireframe : 와이어 프레임 표현으로 메시를 그림.

Shaded Wireframe : 텍스처를 입히고 Wireframe을 오버레이하여 메시를 표시한다.

2D, 조명(Lighting) 및 오디오(Audio) 스위치

2D : Scene을 2D 뷰와 3D 뷰 간에 전환한다. 2D모드에서 카메라는 양의 Z축을 향하고 X축은 오른쪽을, Y축은 위를 가리킨다.

Lighting : Scene View 조명(광원, 오브젝트 셰이딩)을 켜거나 끈다.

Audio : Scene View 오디오 효과를 켜거나 끈다.

Effect Button

skybox : Scene의 배경에 렌더링 되는 스카이박스 텍스처

Fog : 카메라에서 거리가 멀어지면서 뷰가 점차 플랫 컬러로 페이드 된다.

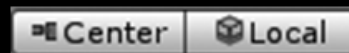
Flares : 광원의 렌즈 플레어

Animated Materials : 애니메이션화된 머티리얼의 애니메이션 표시 여부를 정의한다.



카메라 설정 메뉴: 장면 뷰 카메라를 구성하는 옵션이 있습니다. 이 조정은 GameObjects with Camera 구성 요소의 설정에는 영향을 주지 않습니다.

카메라 설정 메뉴에 액세스하려면 장면 뷰의 도구 모음에서 카메라 아이콘을 클릭합니다.



Position

Pivot : 기즈모의 위치가 해당 메시의 실제 피벗 지점으로 지정.

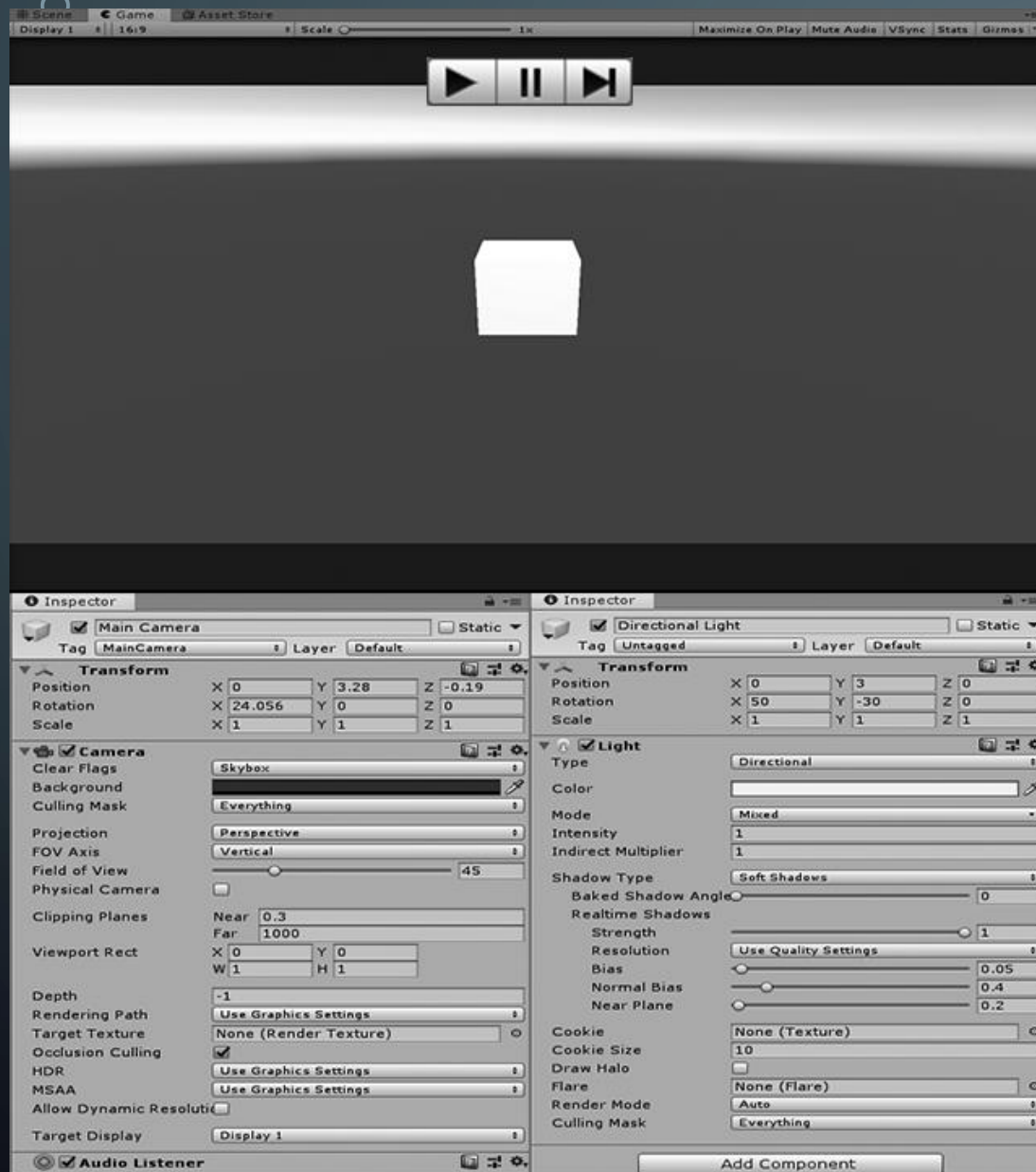
Center : 기즈모의 위치가 게임 오브젝트에 렌더링된 경계의 가운데로 지정.

Rotate

Local : 기즈모가 계속 게임 오브젝트를 기준으로 회전한다.

Global : 기즈모가 월드 공간 방향으로 고정된다.

4. INTERFACE



Game View

Scene View에 배치된 Camera, Light를 중심으로 Camera에 셋팅된 값으로 Scene View의 GameObject들을 렌더링하여 디스플레이에 보여준다.

Unity 가운데 상단에 나와있는 3개의 버튼으로 GameView 실행과 일시 정지를 반복 할 수 있다.

Display1 : 타겟 디스플레이를 설정한다.

Free Aspect : 해상도 설정

Status : 현재 플레이 되고 있는 리소스(재원) 사용량을 보여준다.

Inspector View

Hierarchy나 Project View에 있는 GameObject들의 속성과 Component들을 보여주고 값을 조정하거나 초기값을 셋팅 하는등 Editor상에서 할 속성값 조절을 하게 해주고 GameObject가 필요로 하는 Component들을 링크 시켜준다.