



뉴비유니티 스타트!!

오승준

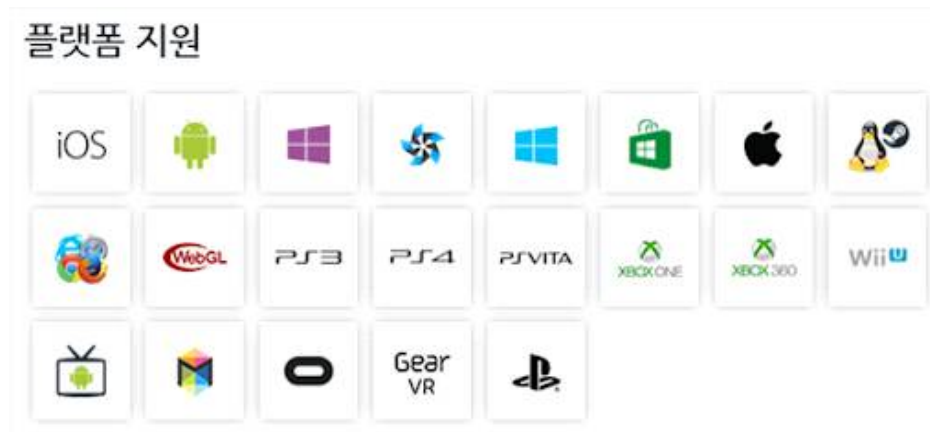


첫 프로젝트 만들기

유니티 프로젝트 만들기

○○○ 유니티로 비디오게임을 만들어요 ○○○

- 유니티를 사용하면 비교적 간단하게 게임을 만들 수 있음.
- 유니티가 비디오 게임 개발의 문턱을 많이 낮춤
- 멀티플랫폼을 지원함



유니티 렌더 파이프 라인





유니티 렌더 파이프라인



- “ Unity에서 다양한 렌더 파이프라인을 선택 할 수 있다.
- “ Unity는 서로 다른 기능과 성능 특성을 가진 세 개의 렌더 파이프라인을 제공한다.





렌더 파이프라인 소개



- “ 렌더 파이프라인은 씬의 콘텐츠를 가져와서 화면에 표시하는 작업을 수행한다.
- “ 수행하는 작업은 다음과 같다.
 - “ 컬링
 - “ 렌더링
 - “ 포스트 프로세싱
- “ 렌더 파이프라인은 저마다 다른 기능과 성능적 특성을 지니고 있다.
- “ 렌더 파이프라인은 서로 다른 셰이더 출력을 사용하고 서로 다른 기능을 갖고 있다.
- “ 때문에 한 렌더 파이프라인에서 다른 렌더 파이프라인으로 프로젝트를 전환하기 어렵다.
- “ 따라서 Unity가 제공하는 다양한 렌더 파이프라인을 이해해야 프로젝트에 대한 올바른 결정을 할 수 있다.



렌더 파이프라인 종류



- “ 빌트인 렌더 파이프라인:
 - “ Unity의 기본 렌더 파이프라인. 범용으로 사용되는 렌더 파이프라인,
 - “ 커스터마이징 옵션이 제한적이다.
- “ 유니버설 렌더 파이프라인(URP):
 - “ 쉽고 빠르게 커스터마이징 할 수 있는 스크립터블 렌더 파이프라인.
 - “ 광범위한 플랫폼에서 최적화된 그래픽스를 구현하도록 지원한다.
- “ 고해상도 렌더 파이프라인(HDRP):
 - “ 스크립터블 렌더 파이프라인.
 - “ 고사양 플랫폼을 위한 최신 고해상도 그래픽스를 구현하도록 지원한다.
- “ Unity의 스크립터블 렌더 파이프라인 API를 사용하여 커스텀 렌더 파이프라인을 생성할 수 있다.



빌트인 렌더 파이프라인



- “ Unity의 빌트인 렌더 파이프라인은 Unity의 예전 렌더 파이프라인으로, 스크립터블 렌더 파이프라인에 기반하지 않는다.
- “ 포워드 레더링
- “ 디퍼드 셰이딩
- “ 레거시 디퍼드
- “ 레거시 버텍스 릿



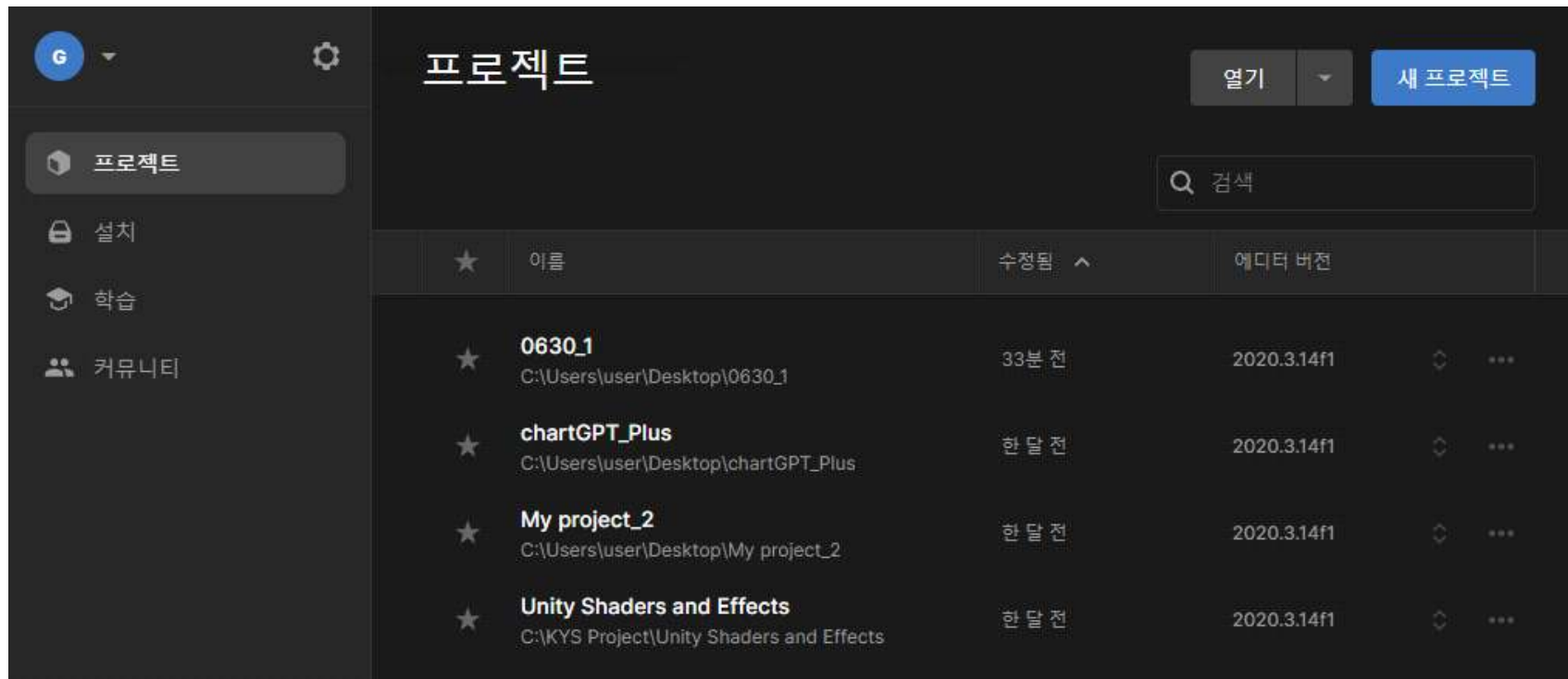
포워드 렌더링



- “ 빌트인 렌더 파이프라인의 기본 렌더링 경로이다.
- “ 이 렌더링 경로는 범용적인 목적으로 사용된다.
- “ 포워드 렌더링으로 실시간 광원을 렌더링하는 작업은 성능 부하를 많이 준다.
- “ 이러한 부하를 줄이기 위해 unity가 픽셀당 한 번에 렌더링하는 광원 수를 선택 할 수 있다.
- “ 썬의 나머지 광원들은 더 낮은 정확도(버텍스 또는 오브젝트의 수)로 렌더링 한다.



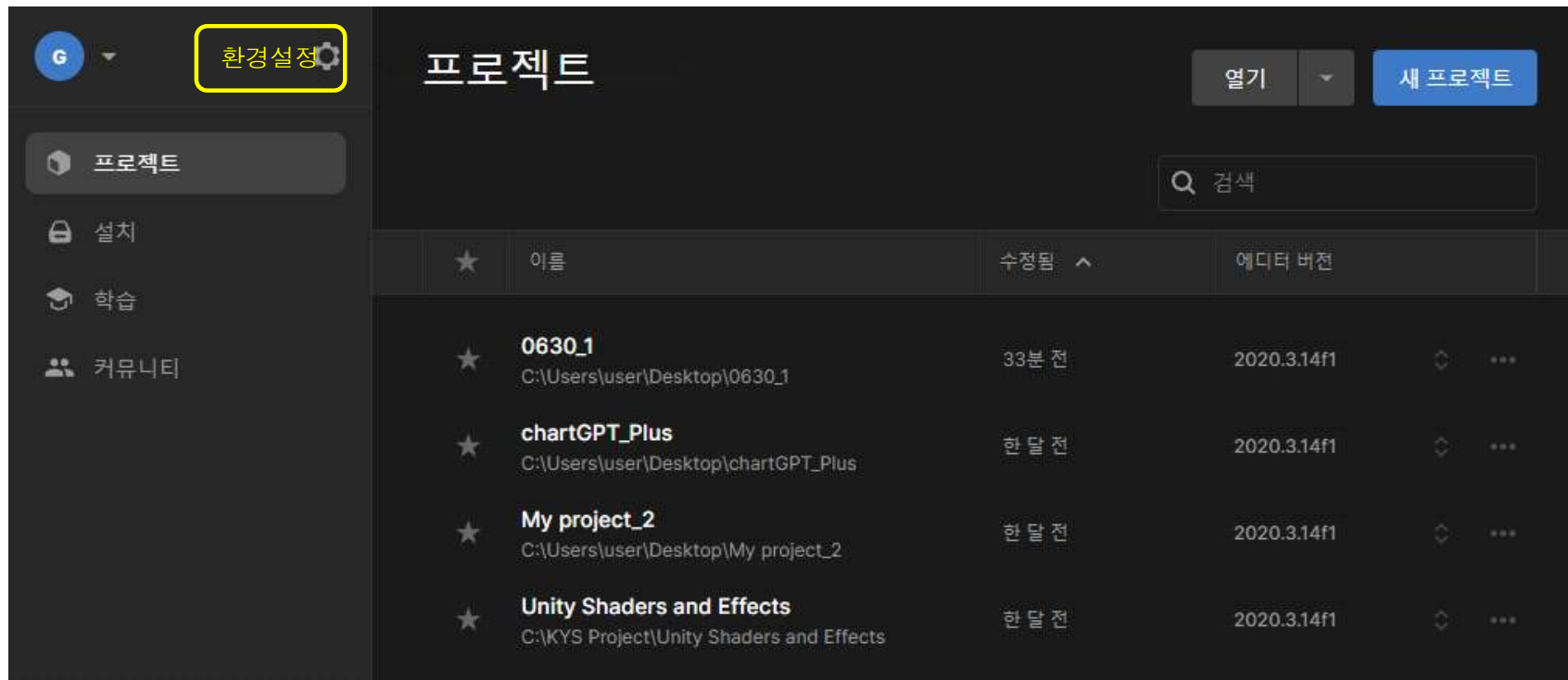
유니티 허브



The screenshot shows the Unity Hub interface. On the left is a sidebar with navigation options: '프로젝트' (Projects), '설치' (Install), '학습' (Learn), and '커뮤니티' (Community). The main area is titled '프로젝트' (Projects) and contains a table of projects. At the top right of the main area are buttons for '열기' (Open), a dropdown arrow, and '새 프로젝트' (New Project). Below these is a search bar labeled '검색' (Search). The table lists four projects, each with a star icon, a name, a path, a last modified time, and an editor version. The projects are: '0630_1', 'chartGPT_Plus', 'My project_2', and 'Unity Shaders and Effects'.

★	이름	수정됨 ^	에디터 버전	
★	0630_1 C:\Users\user\Desktop\0630_1	33분 전	2020.3.14f1	⌵ ...
★	chartGPT_Plus C:\Users\user\Desktop\chartGPT_Plus	한 달 전	2020.3.14f1	⌵ ...
★	My project_2 C:\Users\user\Desktop\My project_2	한 달 전	2020.3.14f1	⌵ ...
★	Unity Shaders and Effects C:\KYS Project\Unity Shaders and Effects	한 달 전	2020.3.14f1	⌵ ...

환경 설정

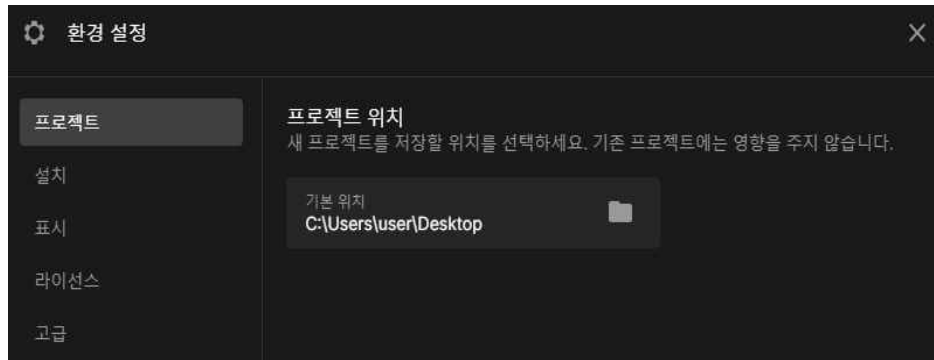


The screenshot shows the Unity Hub interface. In the top left, the '환경설정' (Settings) button is highlighted with a yellow box. The main area is titled '프로젝트' (Projects) and contains a table of projects.

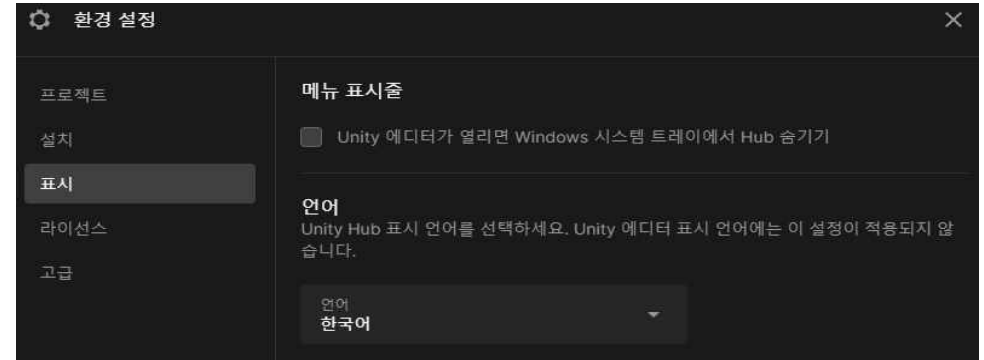
★	이름	수정됨	에디터 버전	
★	0630_1 C:\Users\user\Desktop\0630_1	33분 전	2020.3.14f1	⌵ ...
★	chartGPT_Plus C:\Users\user\Desktop\chartGPT_Plus	한 달 전	2020.3.14f1	⌵ ...
★	My project_2 C:\Users\user\Desktop\My project_2	한 달 전	2020.3.14f1	⌵ ...
★	Unity Shaders and Effects C:\KYS Project\Unity Shaders and Effects	한 달 전	2020.3.14f1	⌵ ...

환경설정 상세 설명

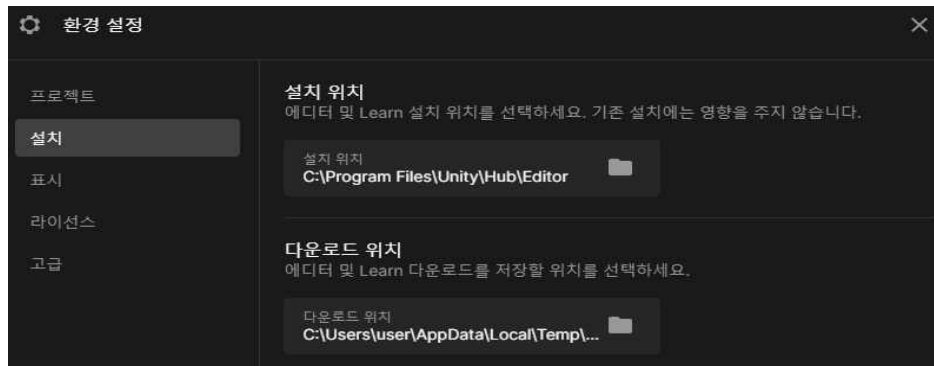
프로젝트



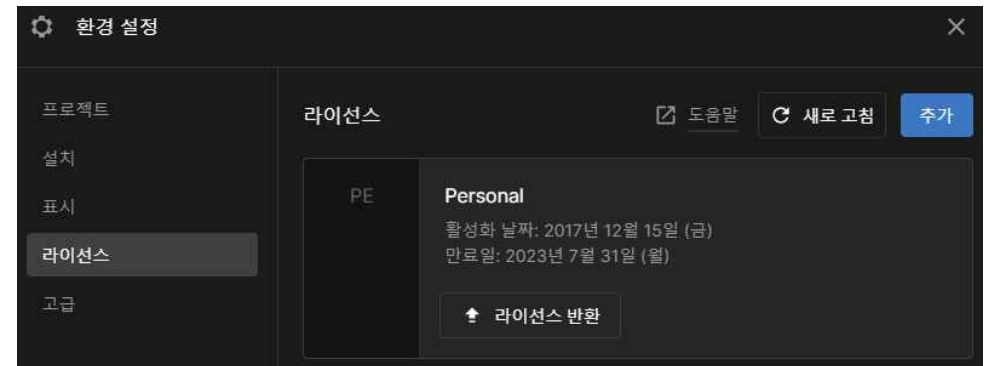
표시



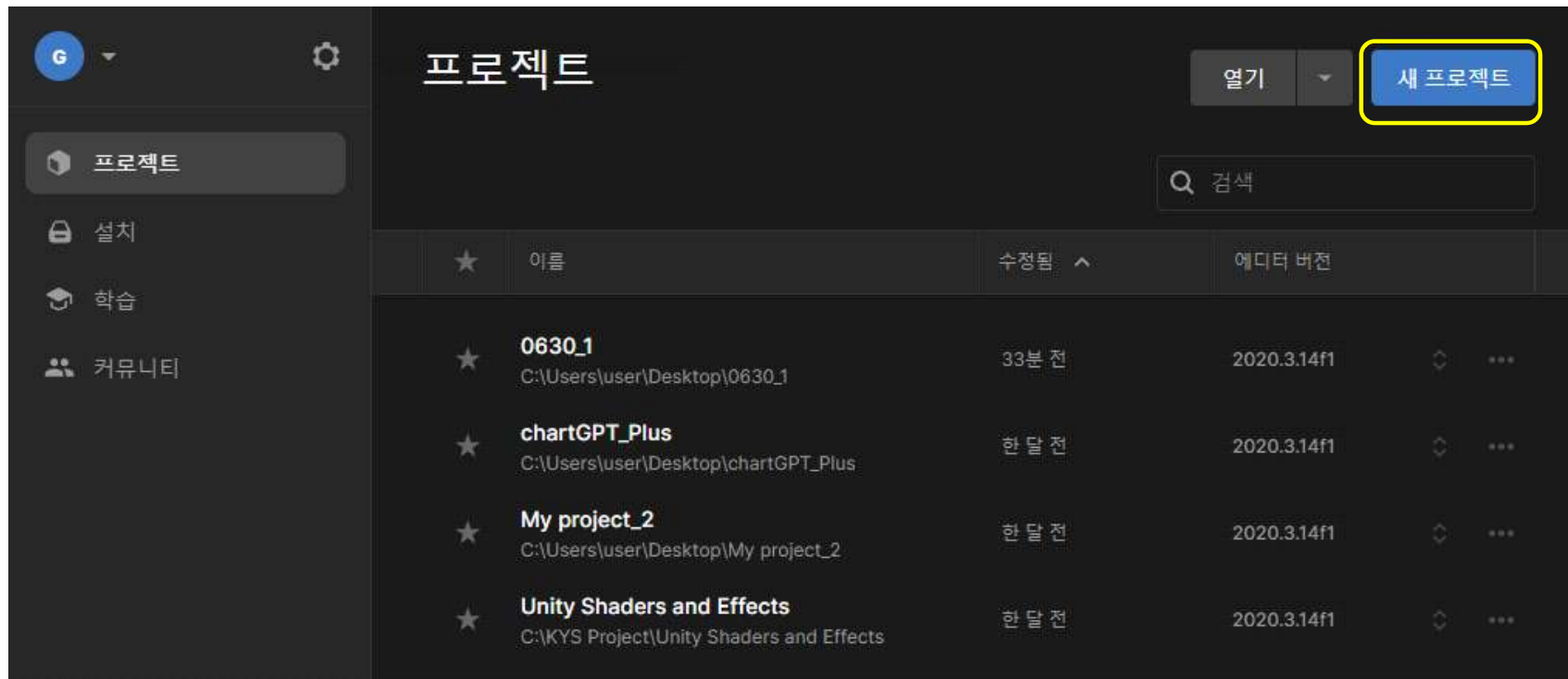
설치



라이선스

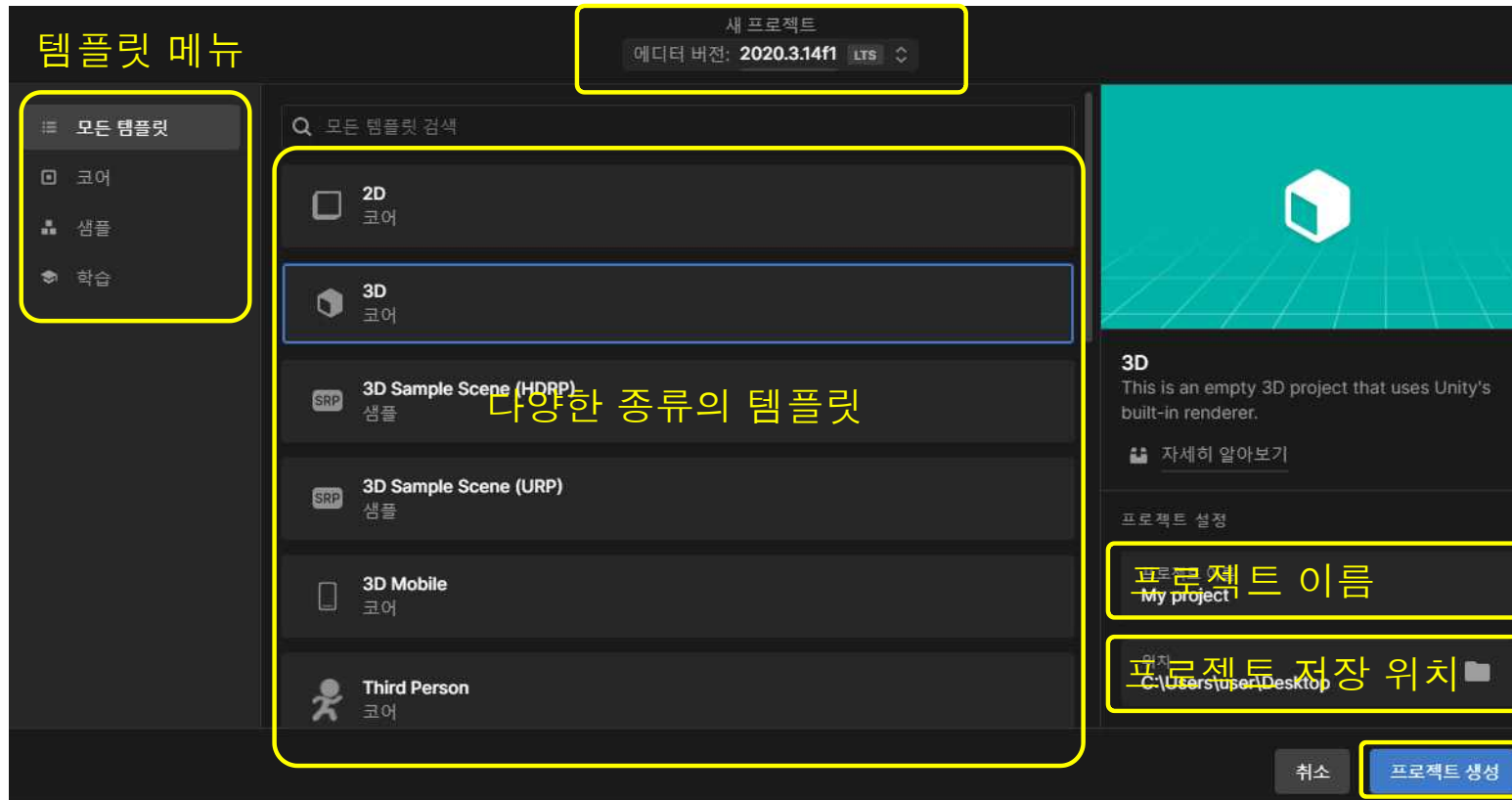


새 프로젝트 만들기



프로젝트 생성

에디터 버전 선택





Unity File Menu



File	Edit	Assets	GameObject	Component	Window
New Scene				Ctrl+N	
Open Scene				Ctrl+O	
Open Recent Scene					>
Save				Ctrl+S	
Save As...				Ctrl+Shift+S	
Save As Scene Template...					
New Project...					
Open Project...					
Save Project					
Build Settings...				Ctrl+Shift+B	
Build And Run				Ctrl+B	
Exit					

새로운 Scene 생성

만들어진 Scene 불러오기

최근 Scene 불러오기

현재 Scene 저장

현재 Scene을 다른 이름으로 저장
(Scene의 이름을 변경하기 위해서는 Save As로 저장해야 함)

새로운 Project 생성

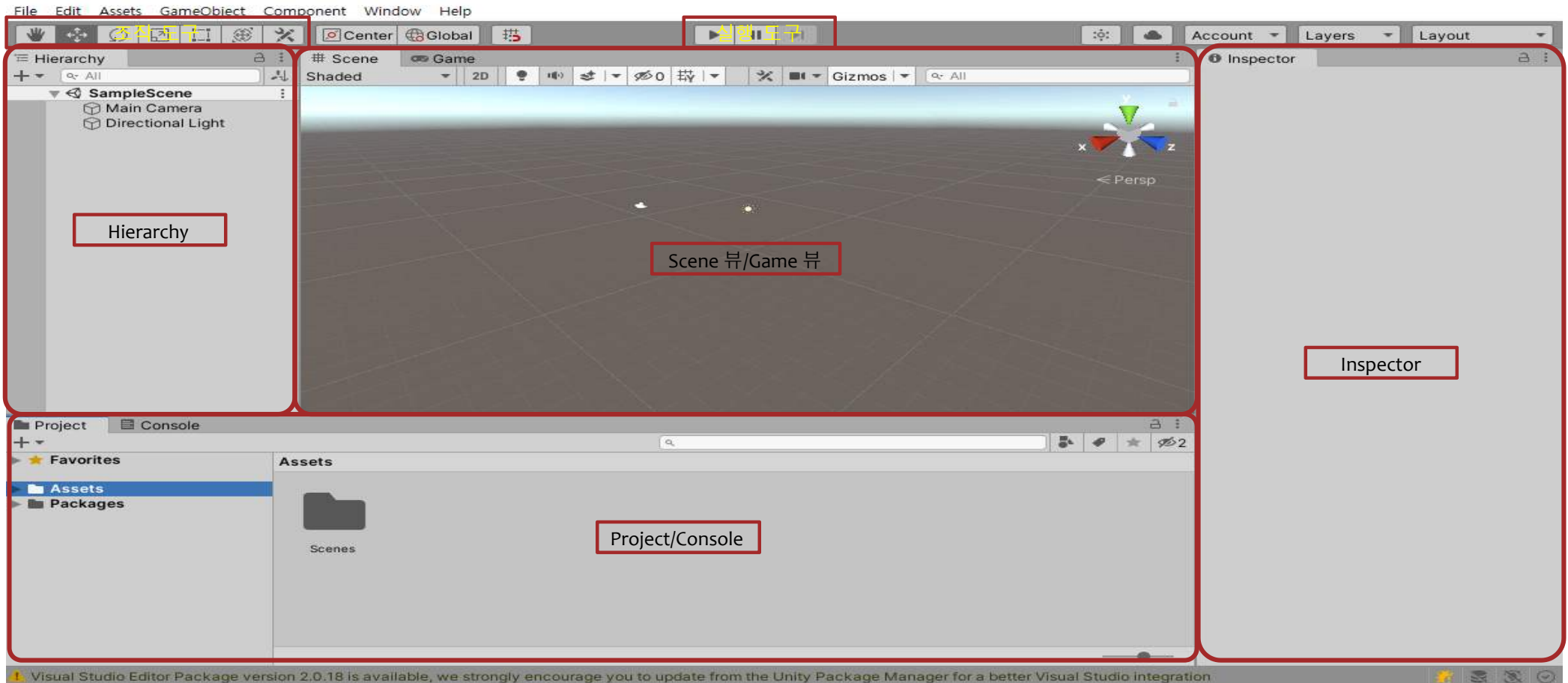
만들어진 Project 불러오기

현재 Project 저장

Build 설정(가능하면 Build를 하기 전에 설정을 하는 것이 좋음)

Build 성공되었다면 자동으로 프로젝트 실행

유니티 화면 설명





유니티 화면 설명



•Scene 뷰

- 게임을 구성하는 메인 화면
- 주로 리소스를 배치해 게임 씬을 작성
- 뷰 위쪽의 탭을 클릭하면 Game 뷰 등을 볼 수 있음

•Game 뷰

- 게임을 실행했을 때 보이는 방향을 확인할 수 있음
- 게임의 처리 속도나 과부하가 걸리는 곳 등을 분석할 수 있음

•Hierarchy 창

- Scene 뷰에 배치한 오브젝트 이름을 목록으로 표시
- 오브젝트 사이의 계층 구조를 표시하거나 편집 가능

•Project 창

- 게임에서 사용하는 리소스를 관리함
- 이미지나 음원 등 리소스를 드래그 & 드롭하면 게임 리소스를 추가 할 수 있음



유니티 화면 설명



•Console 창

- 프로그램에 오류가 있을 때 그 내용을 표시함
- 프로그램에 따라 임의의 수치나 문자열을 표시할 수도 있음

•Inspector 창

- Scene 뷰에서 선택한 오브젝트의 상세 정보가 나타남
- Inspector 창에서 오브젝트의 좌표, 회전, 크기(스케일), 색, 모양 등을 설정

•조작 도구

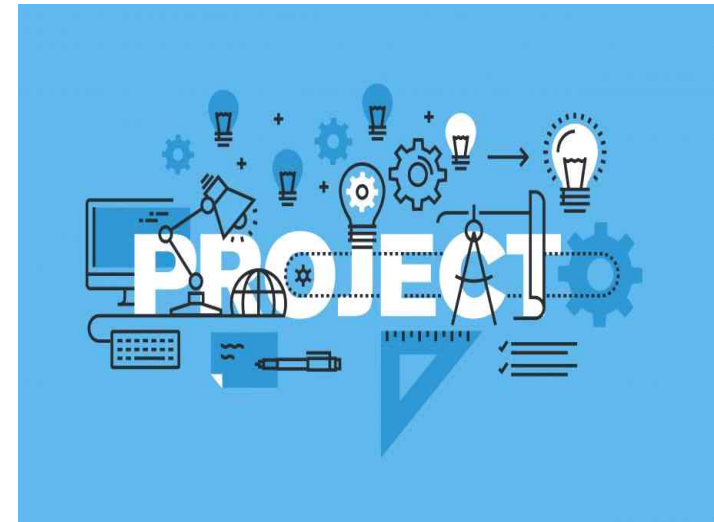
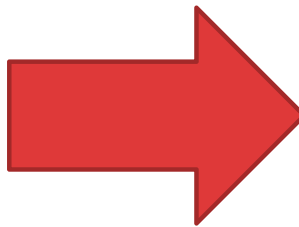
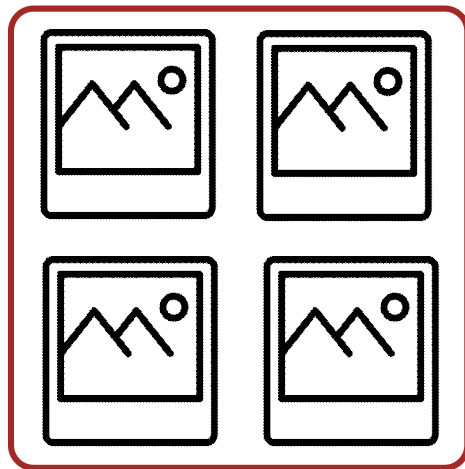
- Scene 뷰에 배치한 오브젝트의 좌표, 회전, 크기를 조절할 수 있음
- Scene 뷰의 보이는 방향을 조정 할 수 있음

•실행 도구

- 게임을 실행하거나 정지시킴

프로젝트 만들기

- 유니티로 게임을 제작하려면 먼저 프로젝트를 만들어야 함
- 유니티에는 '프로젝트'와 '씬'이 있는데, 프로젝트는 게임 전체를 말하고 씬은 장면 단위를 말함
- 연극에 비유하면 각본이 프로젝트, 각 장면이 씬
- '프로젝트=게임'이므로 프로젝트를 만들 때는 게임 타이틀을 프로젝트 이름으로 해 두면 알기 쉬움

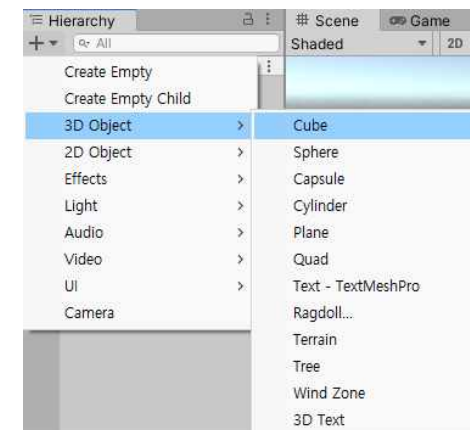
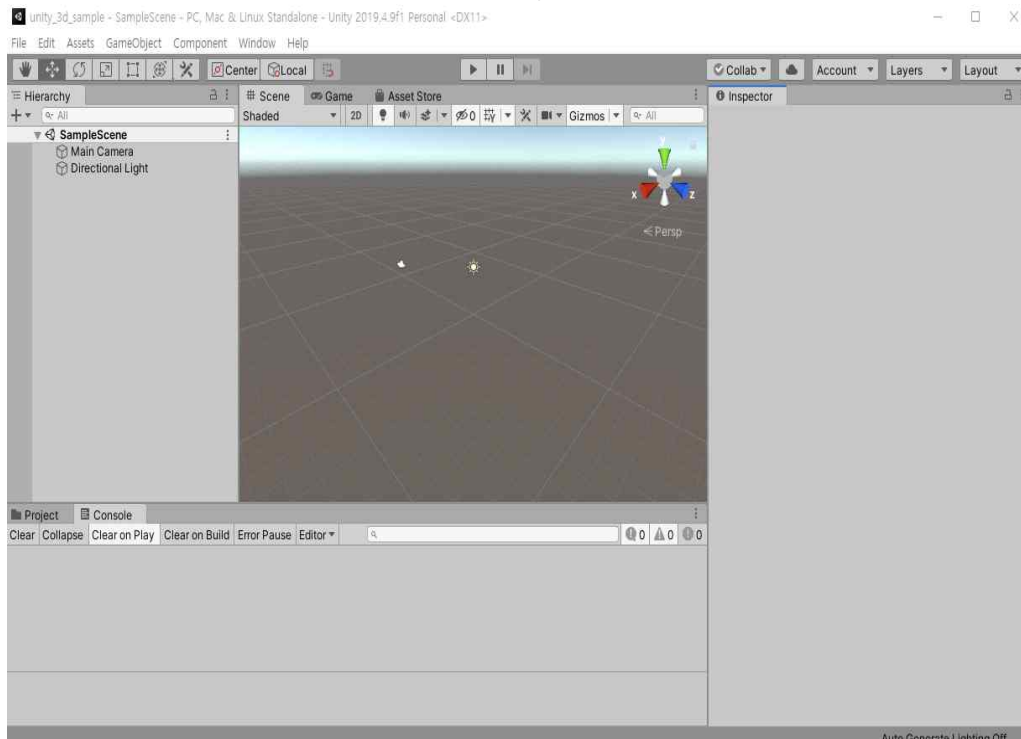




프로젝트 만들기



오브젝트 추가하기(3D Object > cube)

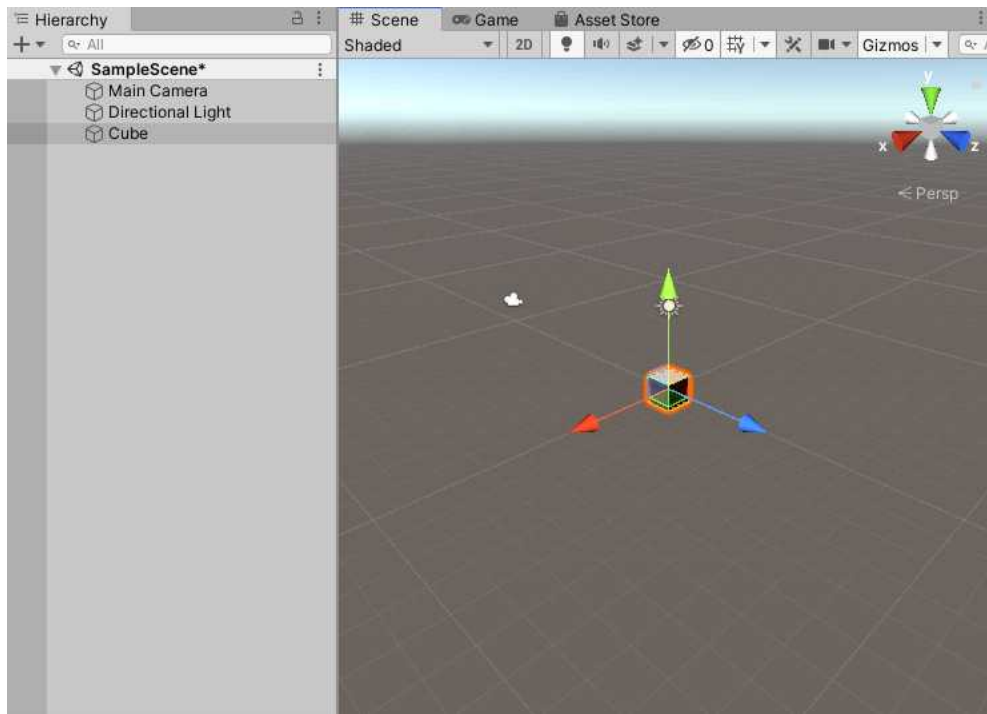




프로젝트 만들기



오브젝트 추가하기(3D Object > cube)

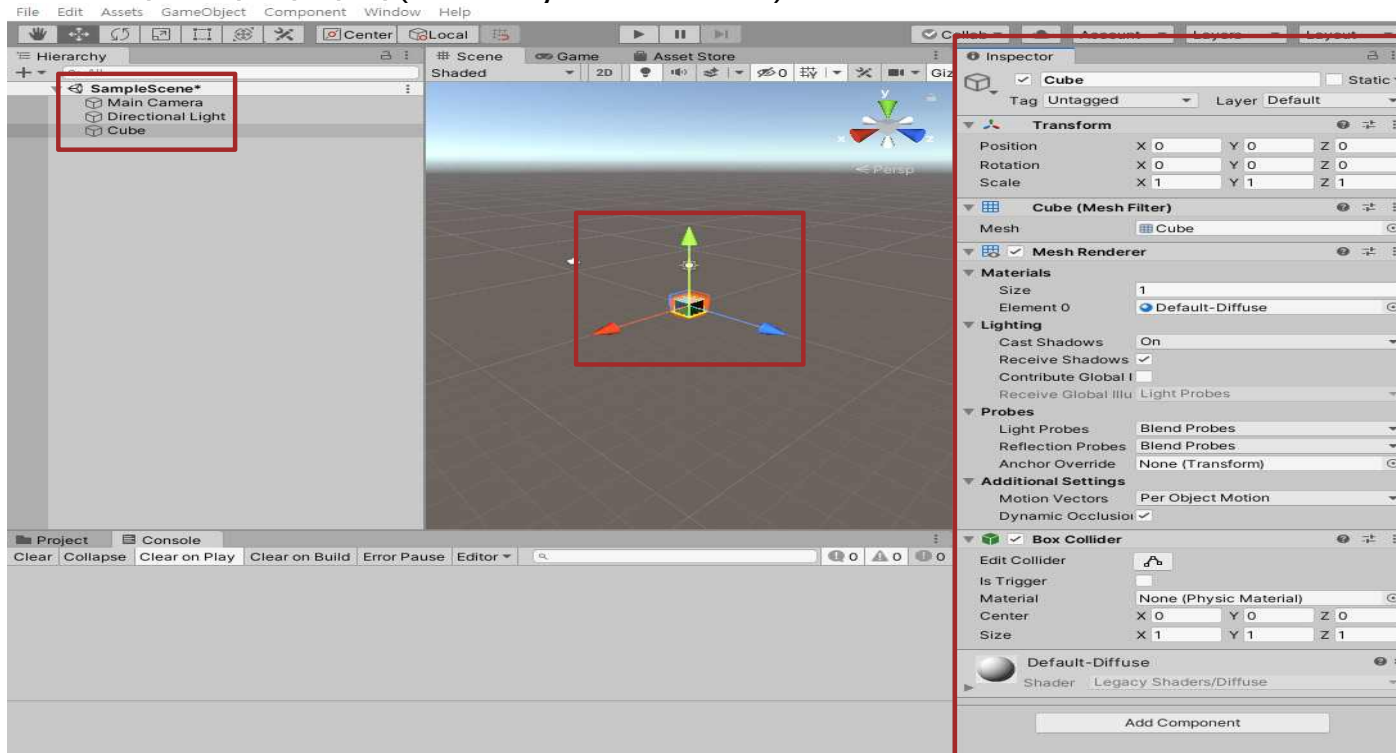




프로젝트 만들기



오브젝트 추가하기(3D Object > cube)

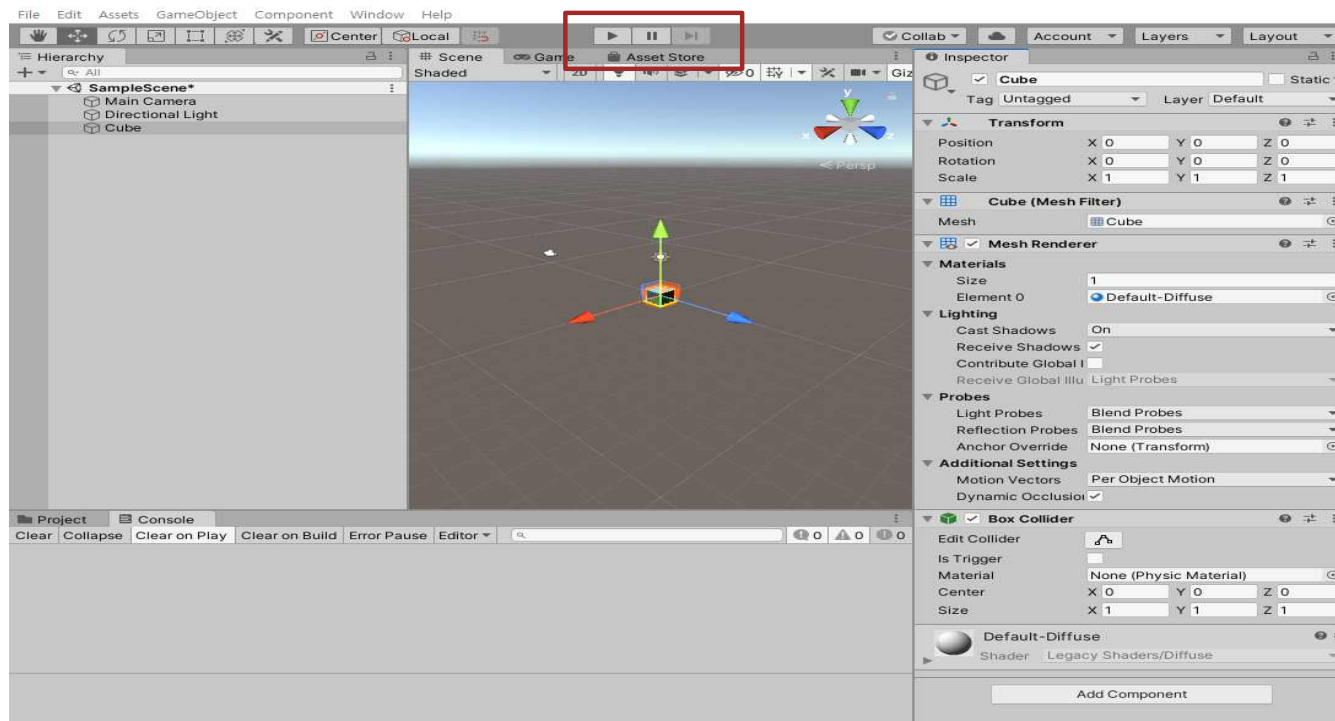




프로젝트 만들기



게임 실행하기

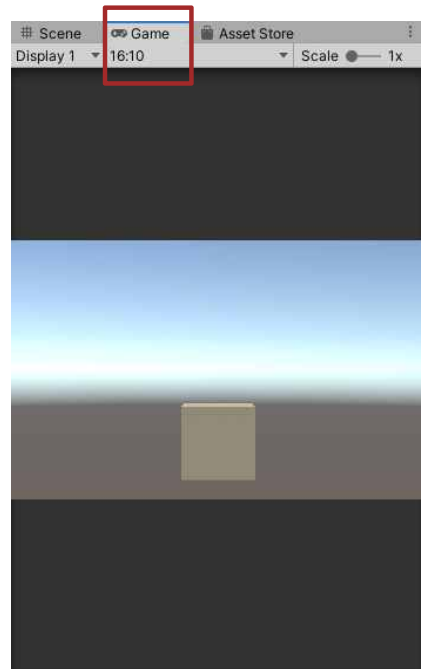
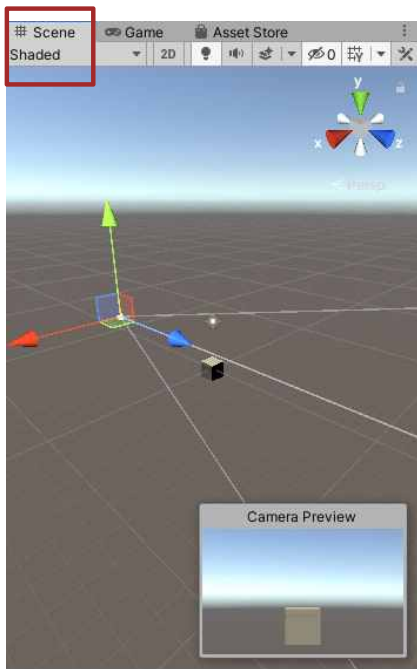




프로젝트 만들기



“ 게임 실행하기(플레이 버튼 선택 후)

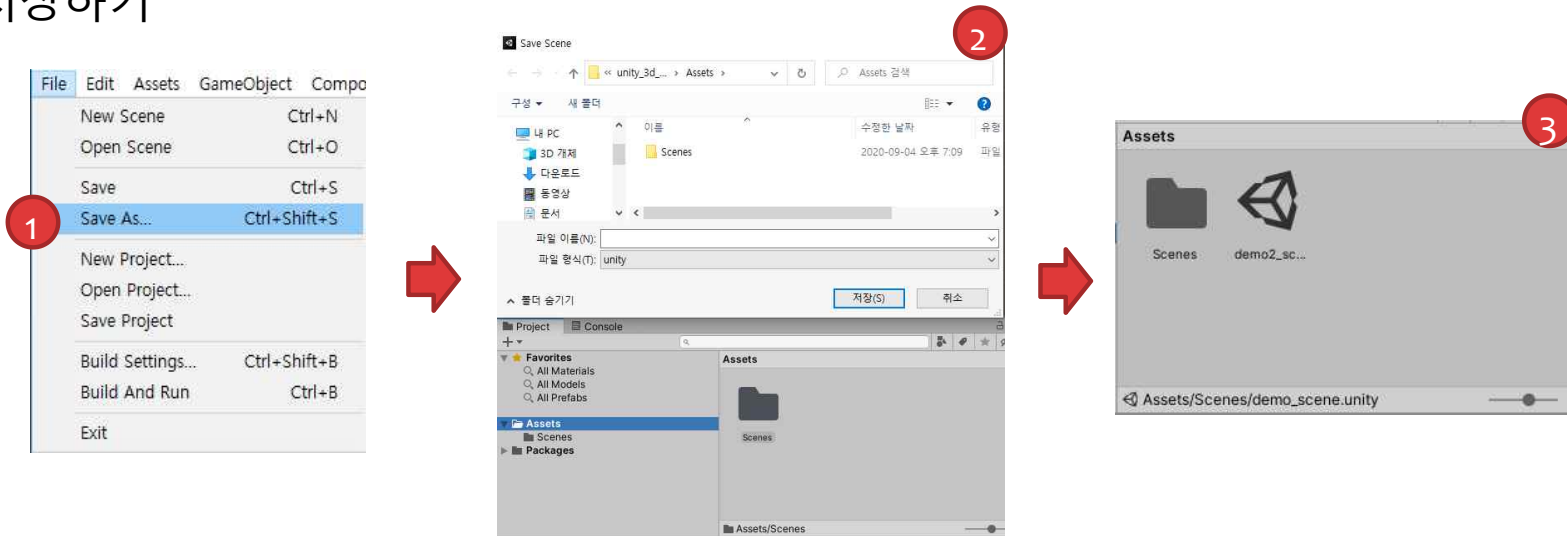




유니티 첫걸음



“ 씬 저장하기





프로젝트 만들기



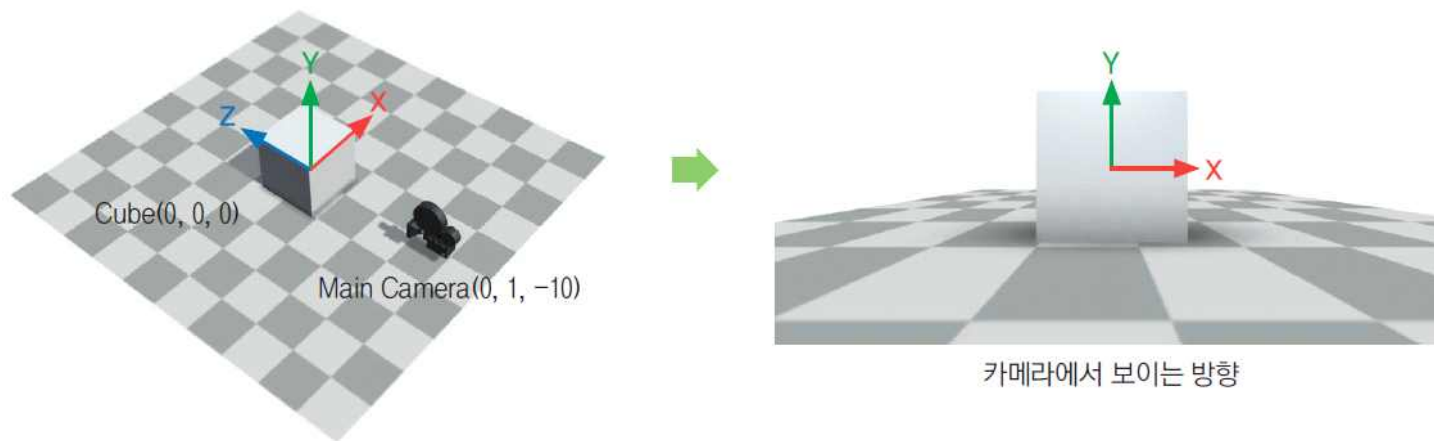
- 오브젝트 추가하기(3D Object > cube)
 - 화면 중앙에 있는 scene 뷰에는 Directional Light 와 Main Camera 아이콘이 있음
 - Inspector 창에서 Transform 항목의 Position을 보면 X, Y, Z 값이 있음
 - X, Y, Z 가 모두 0인 점을 원점이라고 함



유니티 첫걸음



- “ 유니티 3D 좌표계와 보이는 방향의 이미지



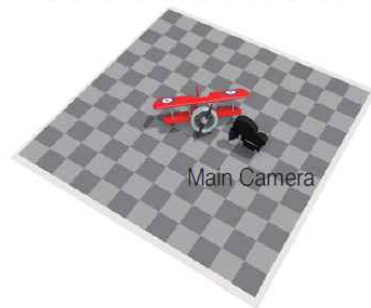


유니티 첫걸음



“ 카메라와 게임 화면의 관계

카메라를 피사체에 가까이 둡니다.



카메라에서 보이는 방향과 크기

카메라를 피사체에서 멀리 떨어뜨립니다.



카메라에서 보이는 방향과 크기



유니티 첫걸음



“ 게임 실행하기

- “ 실행 도구를 한 번 더 클릭하면 게임이 중지되고 Scene 뷰로 되돌아감
- “ 이처럼 카메라로 촬영한 이미지가 게임 실행 화면이 됨
- “ 카메라가 피사체에서 멀어지면 게임 실행 화면에 비치는 피사체는 작아짐
- “ 반대로 카메라가 피사체와 가까워지면 실행 화면에 비치는 피사체는 커짐



유니티 첫걸음



“ 주의할 점

- “ 유니티에서 3D 게임을 만들려면 공간을 파악하는 능력이 매우 중요함
- “ 공간을 파악할 때 표시되는 것이 원점과 카메라 위치임
- “ 게임을 만들기 전에 무엇을 어디에 배치할지 정확하게 정해 두면 좋음



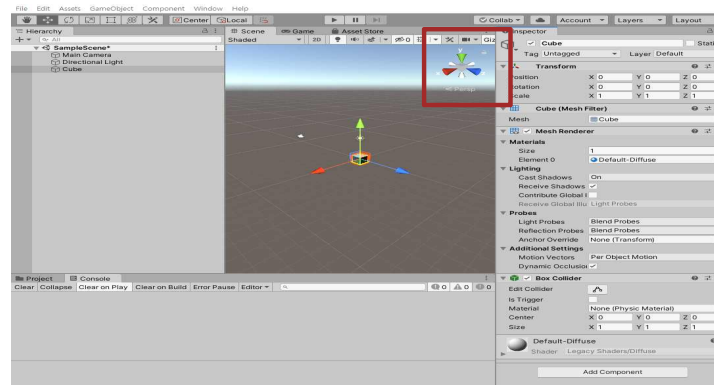
Scene view navigation(화면 탐색)



유니티 첫걸음

“ Hand Tool(Q키)

- “ 마우스 왼쪽 버튼을 누르거나 휠을 누르고 드래그 하면 이동
- “ [Alt]를 누른 채로 화면을 드래그 하면 Scene View에서 시점의 위치가 회전
- “ [Alt]를 누르면 아이콘이 눈 모양으로 바뀜
- “ [Alt]를 누르고 씬이 회전 할 때 화면 오른쪽 위에 있는 씬 기즈모(Scene Gizmo)도 함께 회전
- “ [Ctrl]+[Alt]+왼쪽 마우스 드래그: 씬 뷰의 화면을 상하좌우로 이동(panning)





View Gizmo

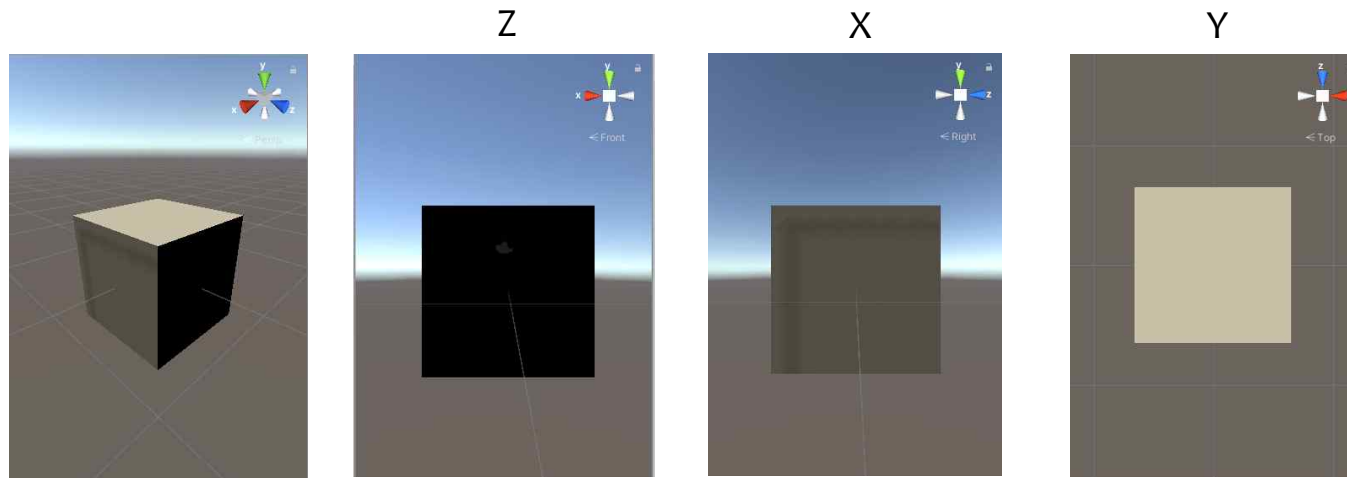


- “ **카메라** 의 현재 방향을 표시하고 시야각과 프로젝션 모드를 변경할 수 있다.
- “ View Gizmo는 큐브의 네 방향에 원뿔 기둥이 있다.
- “ 붉은색 원뿔(**X**) , 초록색 원뿔(**Y**) , 파란색 원뿔(**Z**) 로 레이블이 지정되어 있다 .
- “ 원뿔 기둥 중 하나를 클릭하여 썬 뷰 카메라를 카메라가 나타내는 축(예: 평면도, 왼쪽 뷰 및 전면 뷰)에 스냅 한다.
- “ 큐브를 **마우스 오른쪽 버튼**으로 클릭하여 시야각 목록이 있는 메뉴를 볼 수 있다.
- “ 기본 보기 각도로 돌아가려면 장면 기즈모를 **마우스 오른쪽 버튼**으로 클릭하고 **자유** 를 선택합니다 .
- “ **Perspective** On/Off 할 수 있다.
- “ Scene View 의 투영 모드가 **원근(Perspective)** 과 **직교(Orthographic)** (“등각 투영” (isometric)이라고도 함) 사이에서 변경된다.
- “ View Gizmo의 중앙에 있는 큐브 또는 그 아래의 텍스트를 클릭한다.
- “ 직교 뷰에는 원근감이 없으며 원뿔형 축 암 중 하나를 클릭하여 전면, 상단 또는 측면도를 얻을 때 유용하다.

유니티 첫걸음

“ Scene 뷰에서 시점 조작하기

- “ 씬 기즈모의 빨간색 원뿔을 클릭하면 바로 옆(X 방향)에서 바라본 시점으로 이동
- “ 파란색 원뿔을 클릭하면 정면(Z 방향)에서 바라본 시점으로 이동
- “ 녹색 원뿔을 클릭하면 위(Y 방향)에서 바라본 시점으로 이동
- “ 비스듬한 시점으로 이동하려면 Alt 를 누르면서 드래그해 시점을 회전





화살표 키 사용



- “ **화살표 키**를 사용하여 화면 주위를 이동할 수 있다 .
- “ 위쪽 및 아래쪽 화살표 키는 카메라가 향하는 방향으로 앞뒤로 이동한다.
- “ 왼쪽 및 오른쪽 화살표 키는 보기를 옆으로 이동한다.
- “ 더 빨리 이동하려면 **Shift** 와 **화살표 키**를 누르면 된다 .



View Tool 사용하기



Control	Description
Pan	 클릭하고 드래그하여 카메라를 패닝합니다.
Orbit	 Alt (Windows) 또는 Option (macOS)을 누른 상태에서 왼쪽 클릭하고 드래그하여 현재 피벗점을 중심으로 카메라 궤도를 돌립니다. 보기가 직교이기 때문에 2D 모드에서는 이 옵션을 사용할 수 없습니다.
Zoom	 Alt (Windows) 또는 Option (macOS)을 누른 상태에서 마우스 오른쪽 버튼을 클릭하고 드래그하여 장면 보기를 확대/축소합니다. macOS에서는 Ctrl 키를 누른 상태에서 대신 왼쪽 클릭하고 드래그할 수도 있습니다.



Flythrough 모드



- “ Flythrough모드는 원근 모드 용으로 설계되었다 .
- “ 직교 모드 에서 마우스 오른쪽 버튼을 클릭한 상태에서 마우스를 움직이면 View 대신 카메라를 선회한다.
- “ Flythrough 모드는 2D 모드에서 사용할 수 없다.
- “ 2D 모드에서 **마우스 오른쪽 버튼**을 클릭한 상태에서 마우스를 움직이면 장면 보기 주위로 보기가 Panning을 한다.

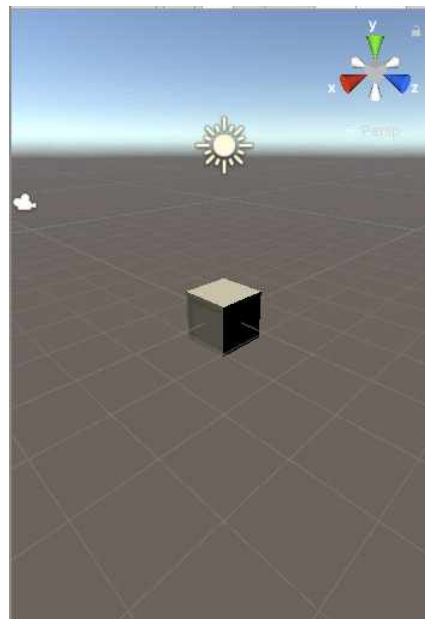
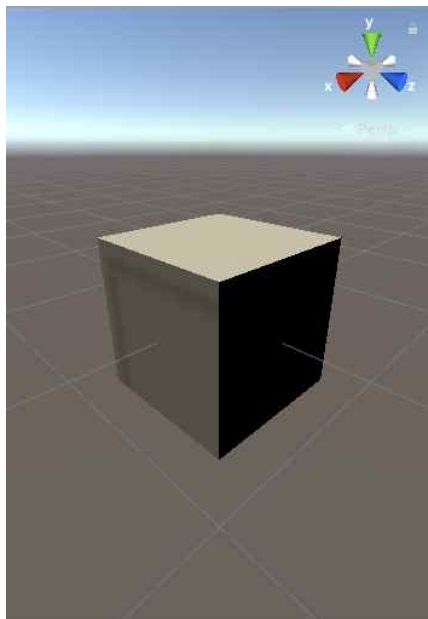
- “ Flythrough 모드를 시작 하고 Flythrough 모드 에서 Scene View를 탐색하려면 다음과 같다.
- “ **마우스 오른쪽 버튼**을 클릭한 상태로 유지한다.
- “ Scene View를 탐색하는 방법은 다음과 같다.
 - “ 마우스를 사용하여 보기를 이동합니다.
 - “ 앞으로 또는 뒤로 이동하려면 W 또는 S를 누릅니다 .
 - “ 왼쪽 또는 오른쪽으로 이동하려면 A 또는 D를 누릅니다 .
 - “ 위 또는 아래로 이동하려면 Q 또는 E를 누릅니다 .
 - “ 더 빨리 이동하려면 Shift 를 길게 누릅니다 .



유니티 첫걸음

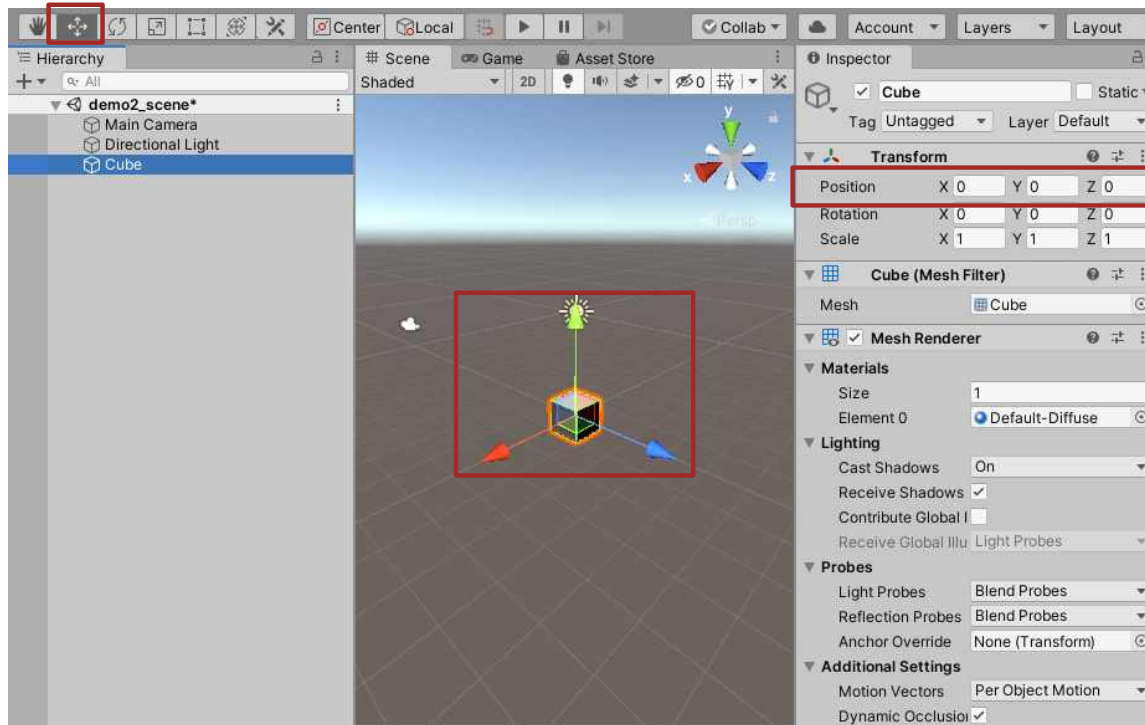


- “ Scene View에서 Zoom In & Zoom Out



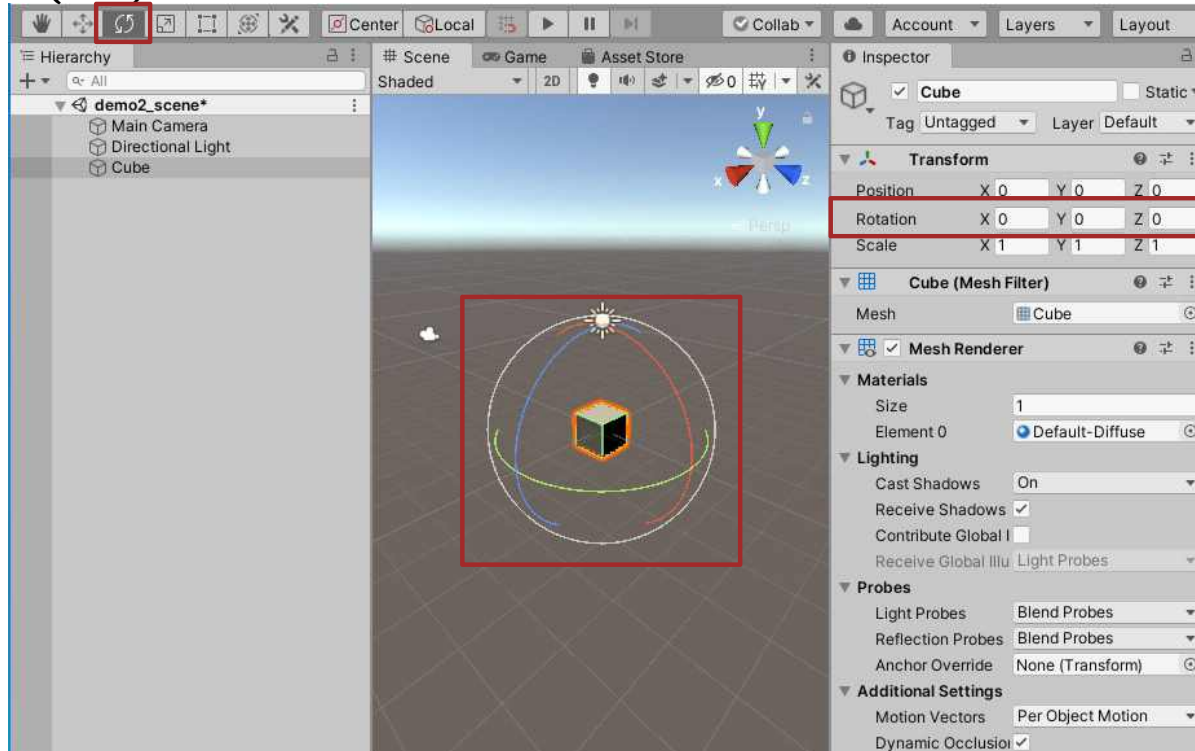
유니티 첫걸음

“ Move Tool(W키)



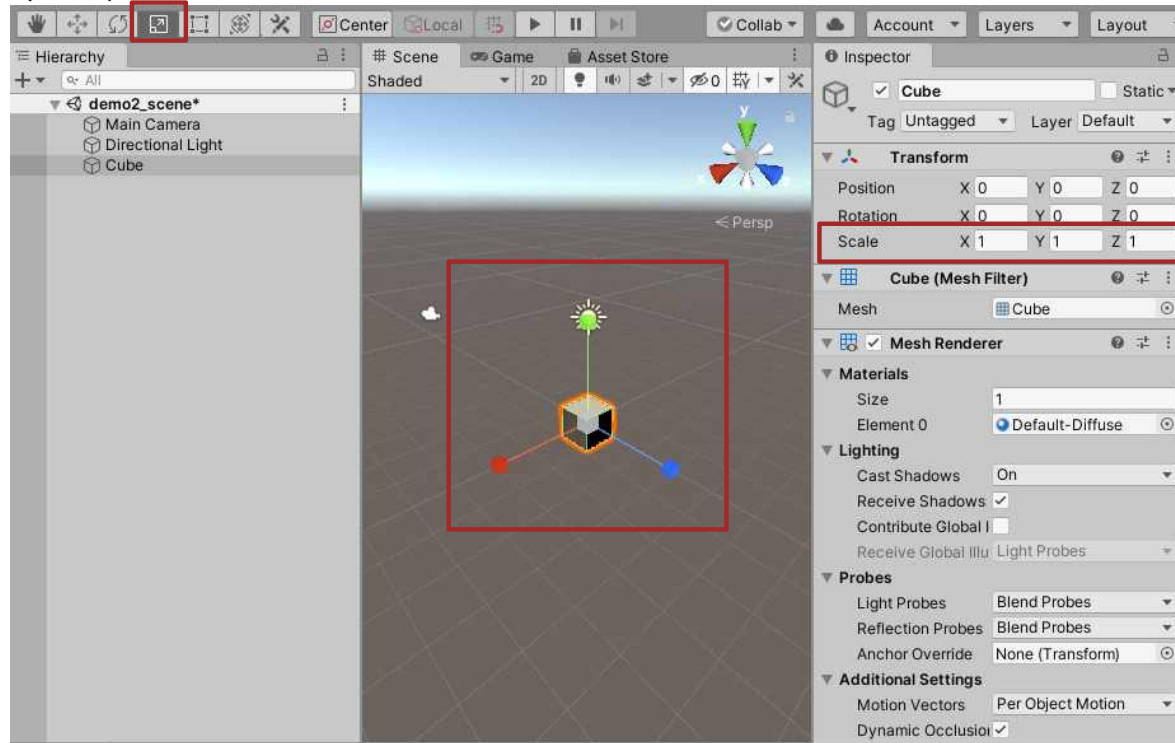
유니티 첫걸음

“ Rotate Tool(E키)



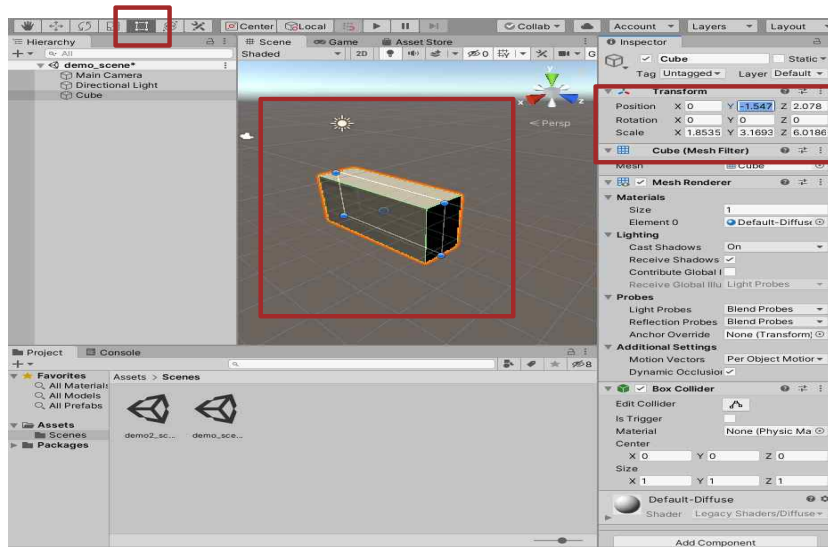
유니티 첫걸음

“ Scale Tool(라키)



유니티 첫걸음

- “ Rect Tool(T키)
 - “ 게임 오브젝트의 크기를 변경하는 툴
 - “ Scale Tool과 다른 점은 크기가 한쪽으로 변하는 점
 - “ 일반적으로 UI, 스프라이트 이미지와 같은 2D 오브젝트를 조정 할 때 사용





유니티 첫걸음



“ Vertex Snapping

- “ [Ctrl] + D키 를 누르면 큐브와 똑같은 큐브를 복제함
- “ 일반적으로 UI, 스프라이트 이미지와 같은 2D 오브젝트를 조정 할 때 사용
- “ 오브젝트를 정확히 붙일 수 있는 기능이 있음
- “ [V]키를 누름
- “ 이것을 Vertex Snapping 이라 부름



유니티 첫걸음

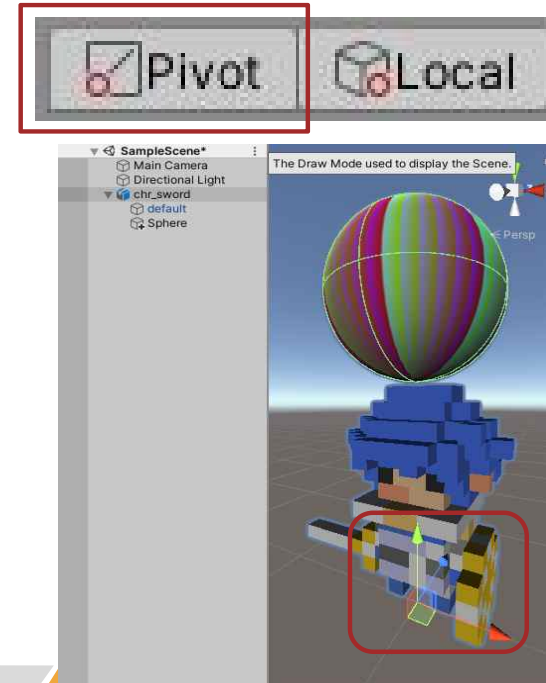
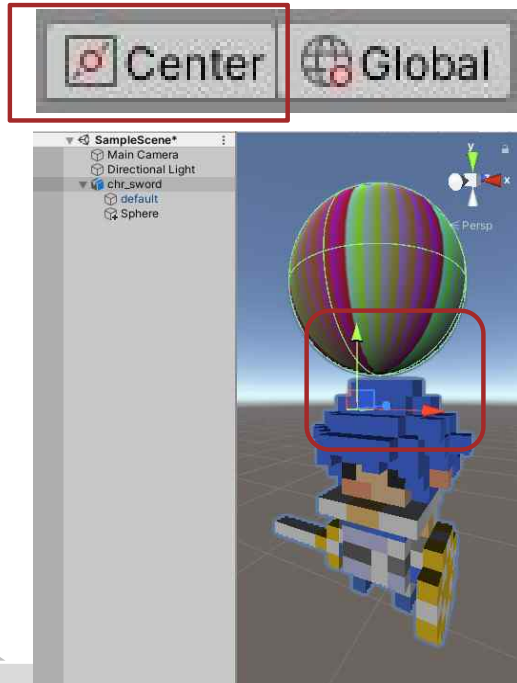


- “ 마우스나 키보드의 조작을 조합함으로써 Scene 뷰의 시점을 조정할 수 있음
 - “ 평행이동
 - “ 마우스 가운데 버튼을 클릭 & 드래그
 - “ Alt 키 + Ctrl 키를 누른 채 클릭 & 드래그
 - “ 회전
 - “ Alt 키를 누른 채 클릭 & 드래그
 - “ 마우스 오른쪽 버튼을 클릭 & 드래그(투영모드가 Perspective 모드라면 주 시점이 아닌 촬영 시점을 중심으로 회전)
 - “ 줌업, 줌다운
 - “ 마우스 스크롤 휠 조작
 - “ 트랙 패드를 두개의 손가락으로 스와이프
 - “ Alt 키를 누른 상태에서 마우스 오른쪽 버튼을 클릭 & 드래그
 - “ 포커스
 - “ Hierarchy 뷰의 오브젝트 이름을 더블 클릭
 - “ 오브젝트를 선택한 후 Shift + F 키

유니티 첫걸음

“ Center / Pivot

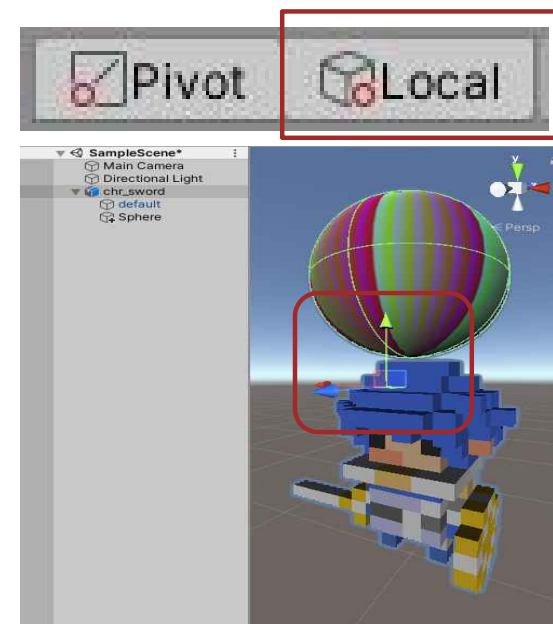
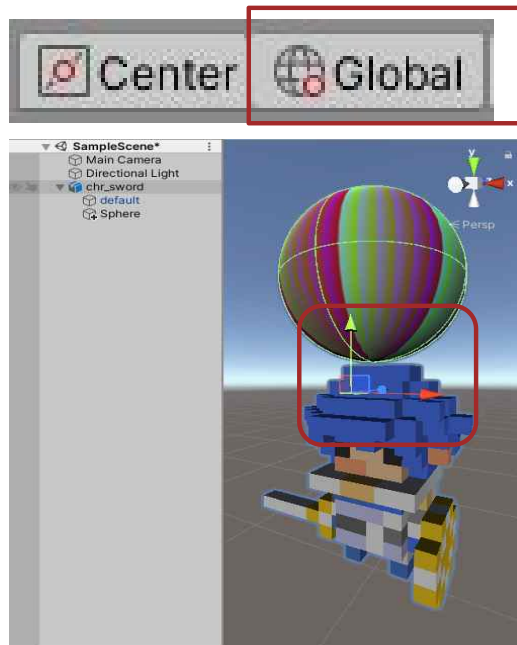
- “ Pivot = 이동 및 회전의 기준점이 부모 오브젝트의 위치
- “ Center = 이동 및 회전의 기준점이 오브젝트의 중간 위치



유니티 첫걸음

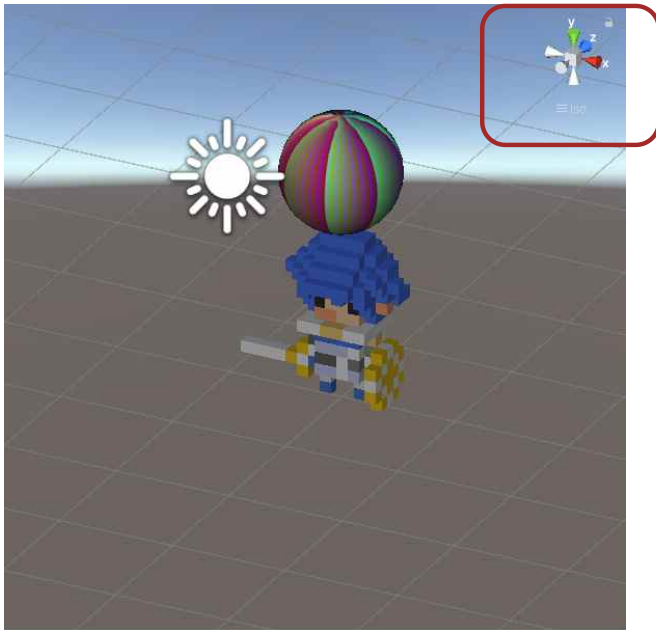
“ Global / Local

- “ Global = 글로벌 표시에서는 오브젝트가 씬의 좌표 축에 따라 이동
- “ Local = 로컬 표시에서는 오브젝트 자신의 방향에 맞춰서 이동



유니티 첫걸음

- “ Iso / Persp
 - “ Iso = 원근감이 없음
 - “ Persp = 원근감이 있음

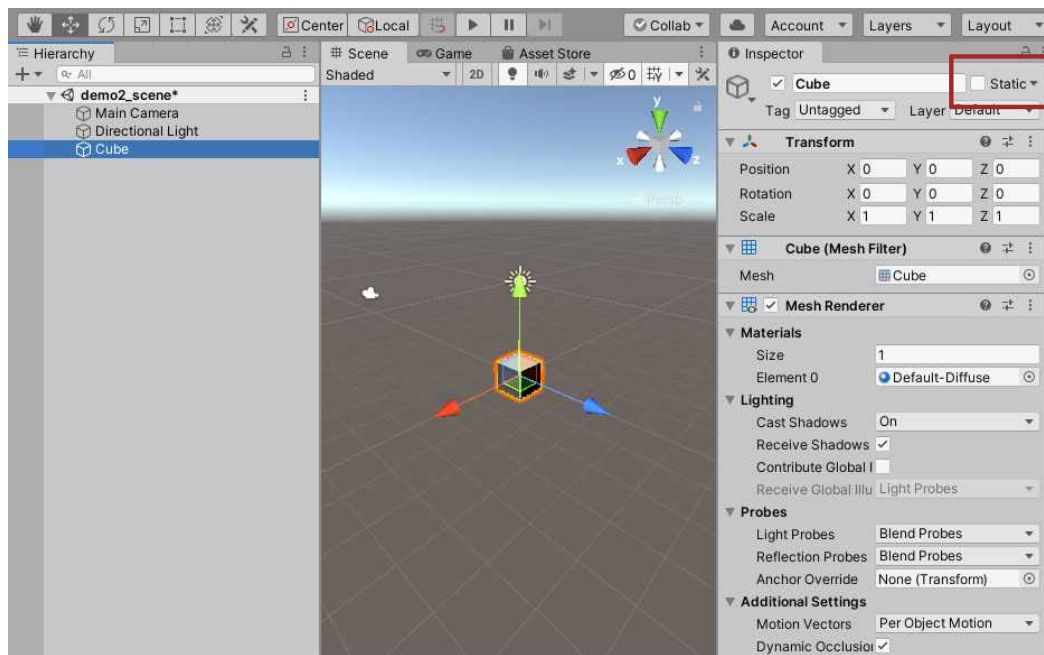




유니티 첫걸음



“ Layout



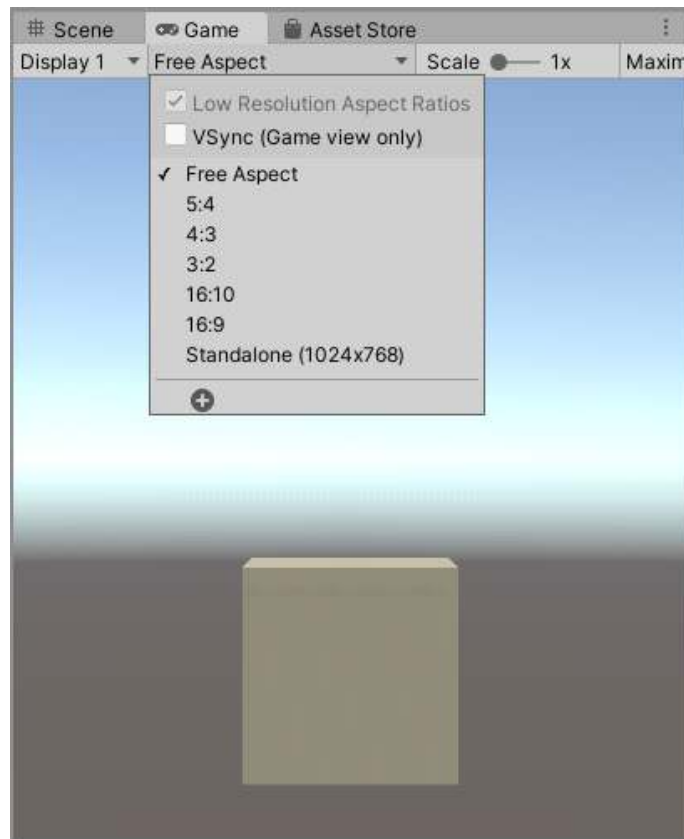
2 by 3
4 Split
Default
Tall
Wide
Save Layout...
Delete Layout...
Revert Factory Settings...



유니티 첫걸음



“ Game View





유니티 첫걸음



“ 유니티 2D 와 3D

- “ 유니티는 3D 뿐 아니라 2D 게임을 만들 수 있는 환경을 제공
- “ 2D게임 와 3D게임은 시점의 차이가 있음
- “ 3D 장면을 정면이나 측면에서 바라보는 시점
- “ 2D는 오브젝트가 카메라에서 떨어져 있어도 오브젝트의 크기가 변하지 않음
- “ 2D는 라이트 효과가 없음

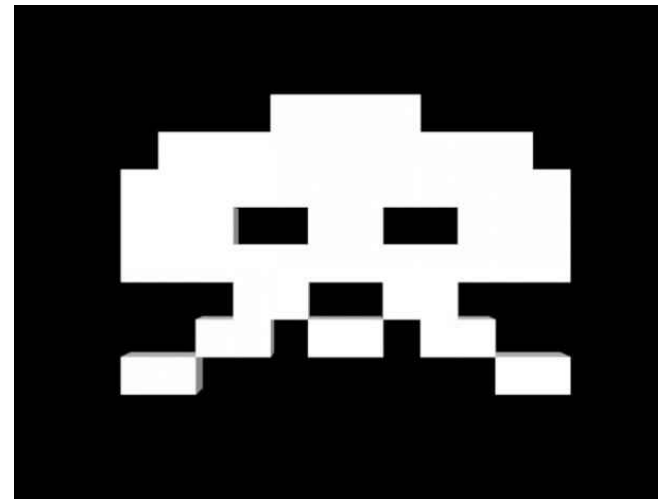
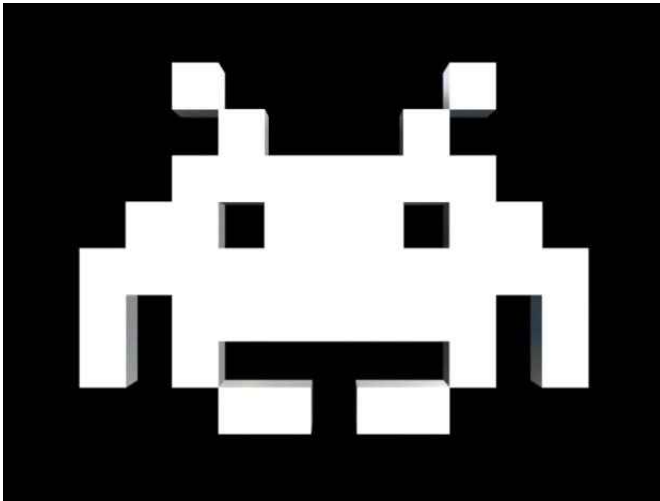




실습 / 과제

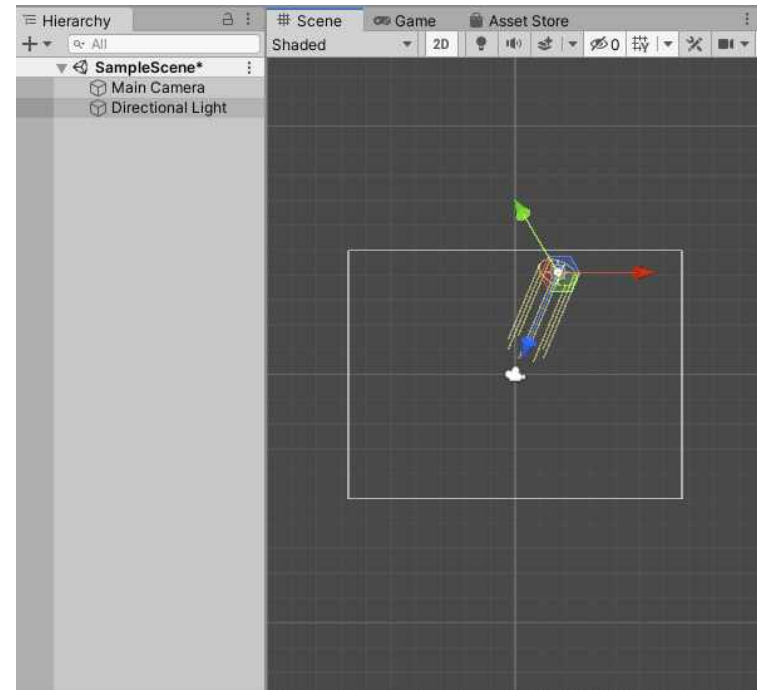
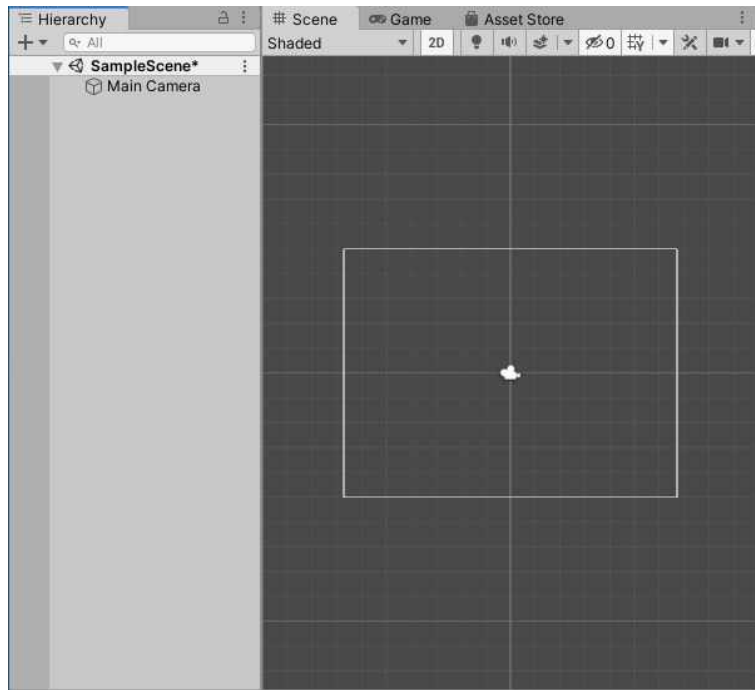


“ 3D 오브젝트를 만들어 봅시다.



○○○ 2D템플릿릿과 3D템플릿릿의 차이 ○○○

Directional Light 의 유무



참고 서적

- “ 그림으로 이해하고 만들면서 배우는 유니티 교과서(길벗, 기타무리 미나미 지음)
- “ Unity4 입문(홍릉과학출판사, 아사노유이치, 아라카와타쿠야, 모리시게요시 지음)
- “ 유니티4 게임 개발의 정석(에이콘, 이득우 지음)
- “ 유니티5로 만드는 3D/2D 스마트폰 게임 개발(제이펍, 요시야미키토 지음)
- “ 게임 제작으로 배우는 유니티(한빛아카데미, 이종원, 정종필 지음)