유니티3D를 활용한

프로그램 코딩

김태완

Index

1. 게임엔진 개요 및 유니티 설치
2. 저작도구 – Unity 3D
3. **미니 게임 제작**

이번 강의에 포함된 내용

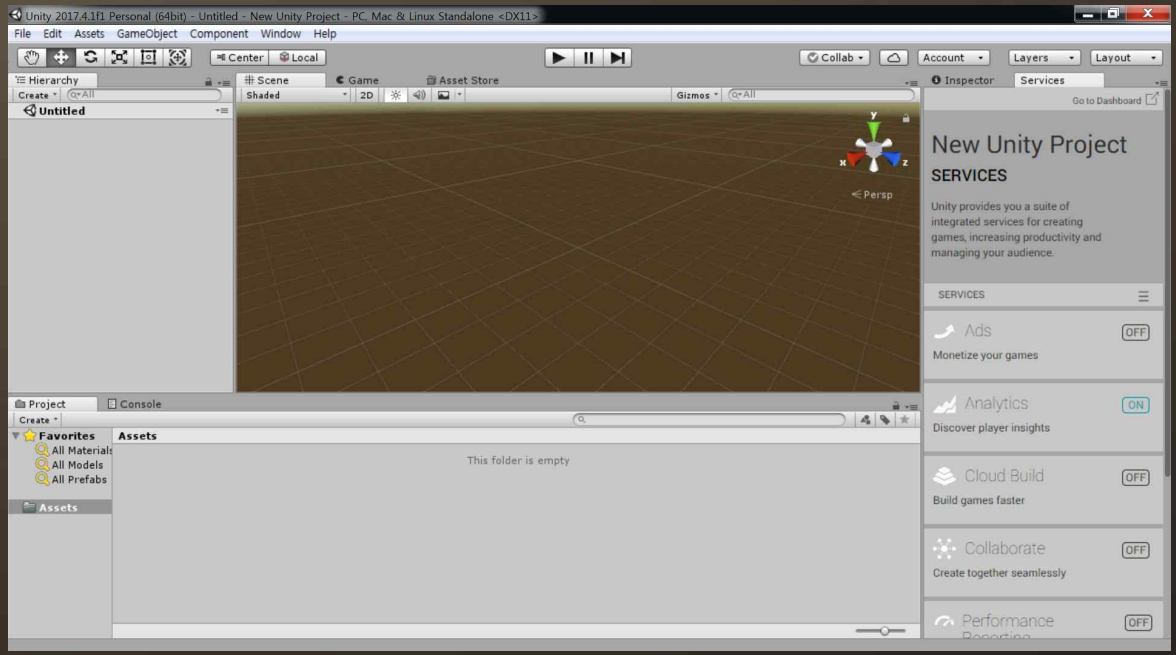
- 미니 게임 제작
  - 간단한 게임을 제작하며 과정을 익히기
- 다양한 시도
  - 미니 게임을 다양한 아이디어로 수정하며 개선

Unity 3D

Mini Game Project(with c#)

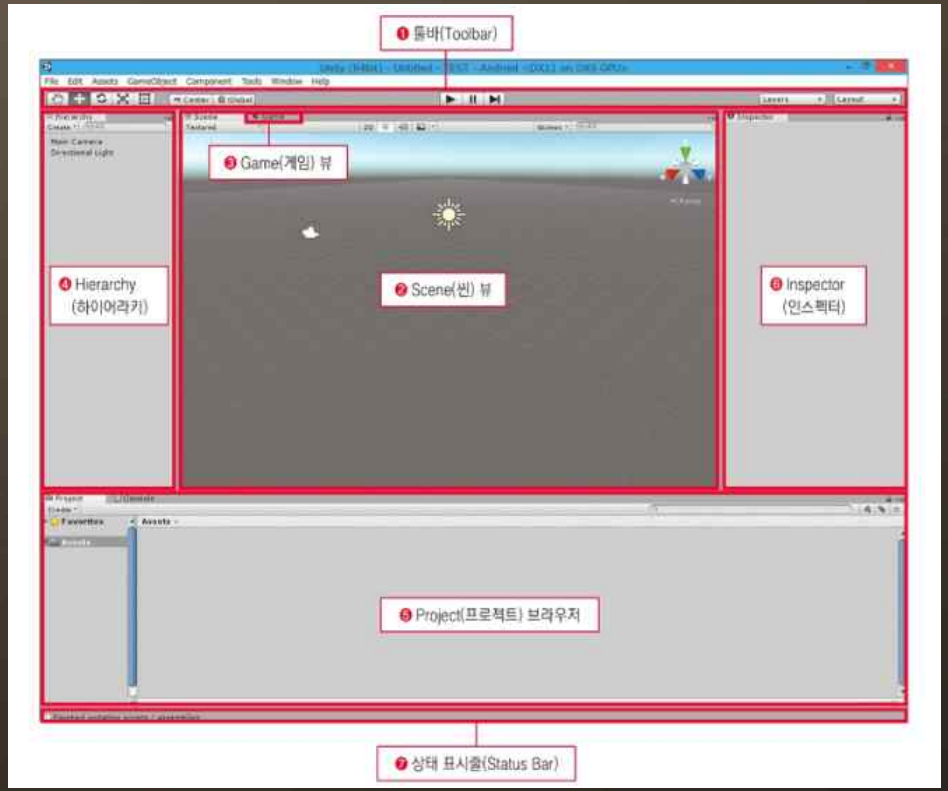
저작도구 - Unity 3D

Unity 3D – 실행화면(기본화면)



창색 변경 : Preference – General - Skin

저작도구 - Unity 3D



- ❶ 툴바Toolbar
- ❷ Scene(씬) 뷰
- ❸ Game(게임) 뷰
- ❹ Hierarchy(하이어라키)
- ❺ Project(프로젝트) 브라우저
- ❻ Inspector(인스펙터)
- ❼ 상태 표시줄Status Bar

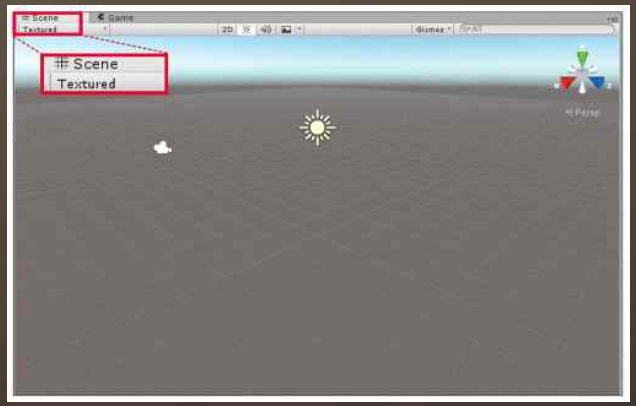
저작도구 - Unity 3D

툴바

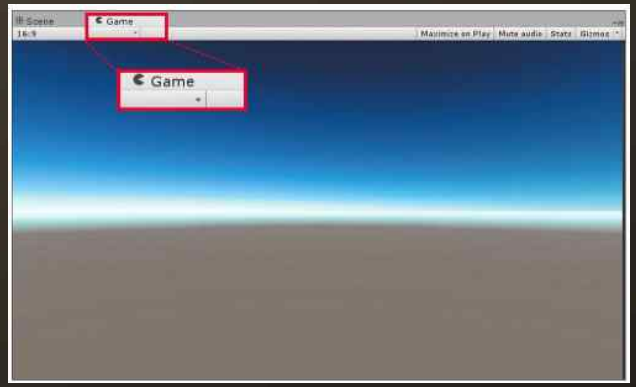
아이콘	이름	내용
	변환 도구(Transform Tools)	가장 왼쪽은 Scene 뷰 안에서 이동하는 버튼(Ctrl 키를 누른 상태에서 드래그하면 화면을 확대·축소할 수 있다), 그 다음은 순서대로 오브젝트 이동 버튼, 오브젝트 회전 버튼, 오브젝트 확대·축소 버튼이다. 왼쪽부터 차례로 Q, W, E, R 키가 단축키로 할당되어 있다.
	변환 기즈모 토글(Transform Gizmo Toggles)	Scene 뷰 안에 있는 기즈모 표시를 전환한다.
	재생/일시 정지/스텝 버튼(Play/Pause/Step Buttons)	Scene 뷰에서 제작 중인 게임을 Game 뷰에서 재생, 일시 정지, 게임을 한 단계씩 끊어서 실행할 때 사용되는 버튼
	레이어 드롭다운(Layer drop-down)	Scene 뷰에 표시되는 레이어를 선택한다.
	레이아웃 드롭다운(Layout drop-down)	화면 레이아웃을 전환한다.

저작도구 - Unity 3D

- Scene View – Scene은 X,Y,Z축을 갖는 3D공간을 말하며 캐릭터, 배경, 소품 같은 다양한 오브젝트를 배치



- Game View – 현재 제작 중인 게임을 Game 뷰에서 실제로 플레이



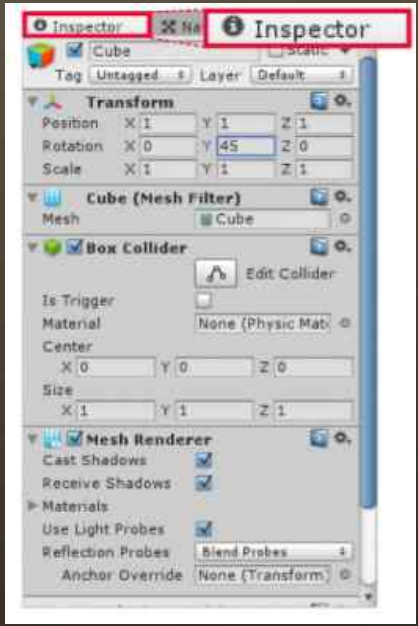
저작도구 - Unity 3D

- Hirerarchy - Scene에 있는 모든 오브젝트가 표시
  게임 오브젝트를 생성/복제/삭제
  오브젝트끼리 상하 혹은 부모와 자식 관계의 계층 구조 설정 및 해제

- Project Browser - 게임에 사용되는 모델, 텍스처, 사운드, 씬 데이터를 포함
  표시되는 내용은 프로젝트 저장 폴더 안에 있는 Assets 폴더의 내용과 같음

저작도구 - Unity 3D

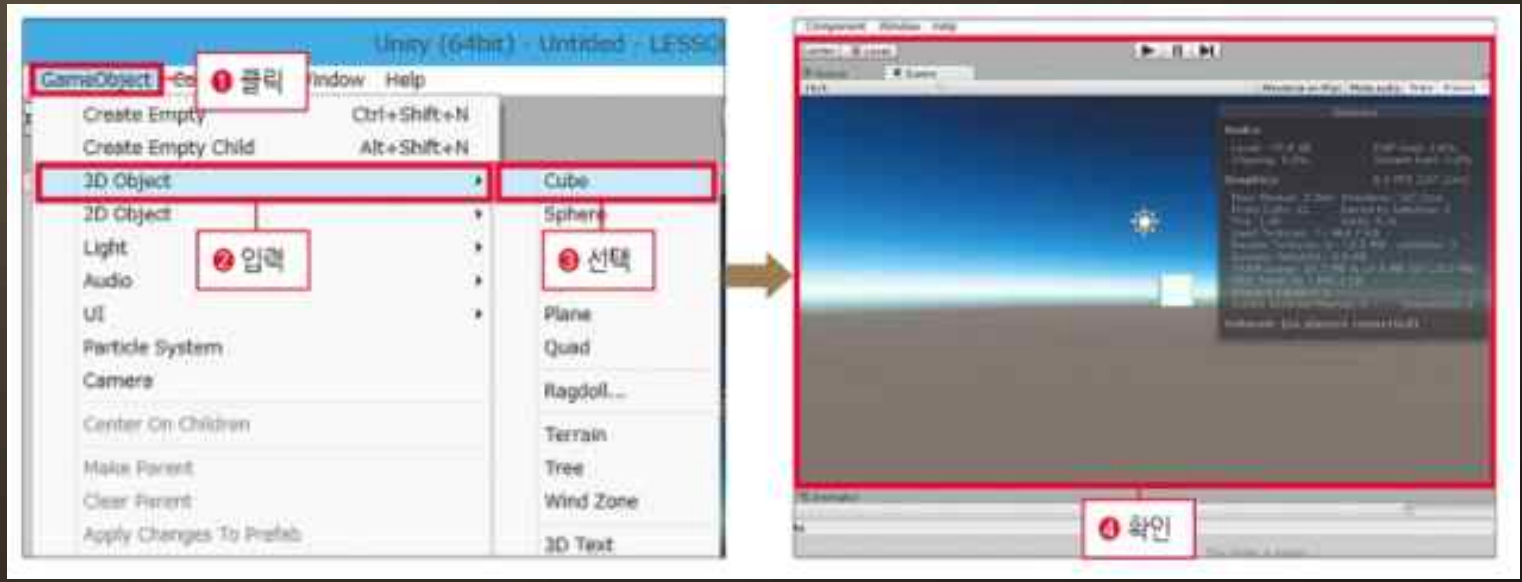
- Inspector – Scene View 또는 Hierarchy에 등록된 오브젝트, Project브라우저의 파일내용 표시하고 위치나 각도를 변경



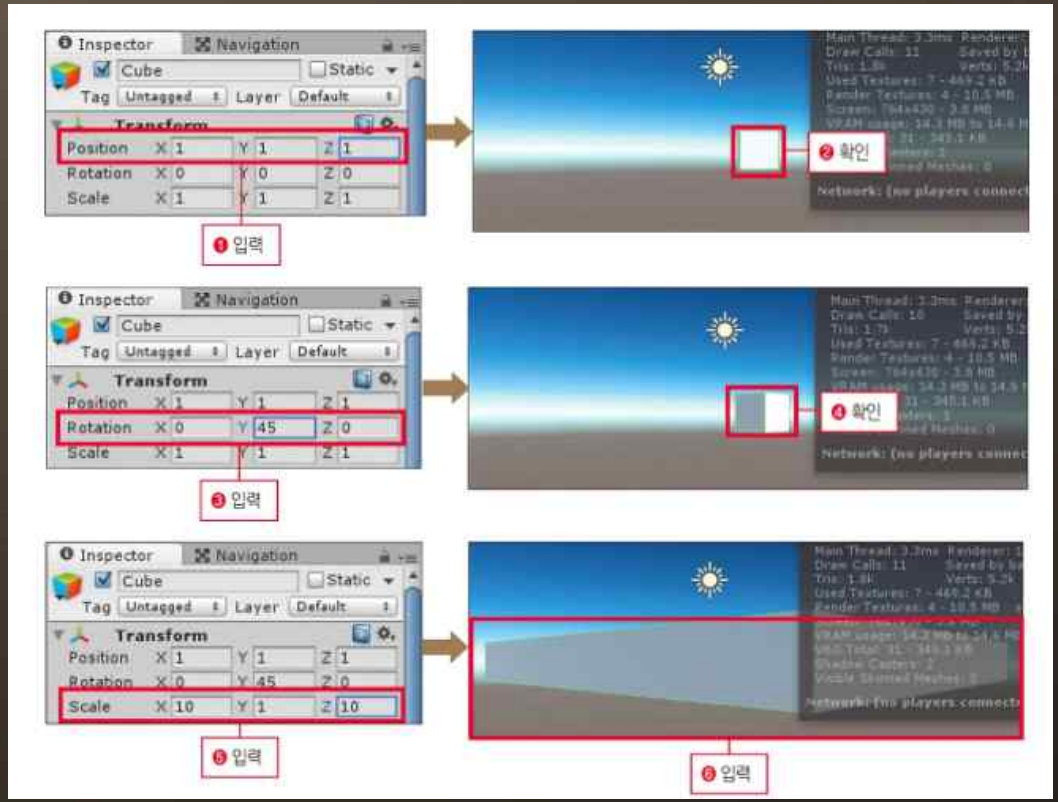
- Status Bar(상태표시줄) – 디버그 메시지나 오류 메시지가 한 줄로 표시

저작도구 - Unity 3D

- 오브젝트 만들기

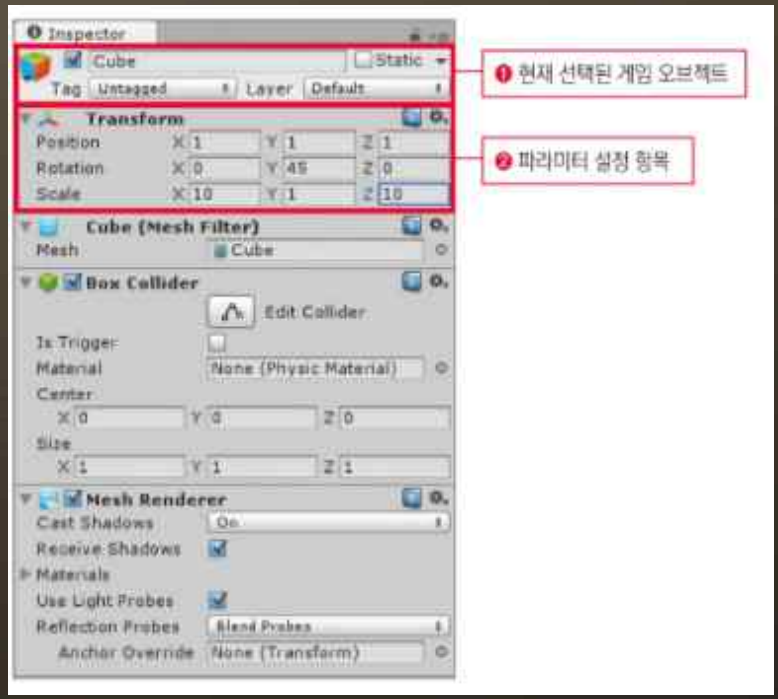


저작도구 - Unity 3D



저작도구 - Unity 3D

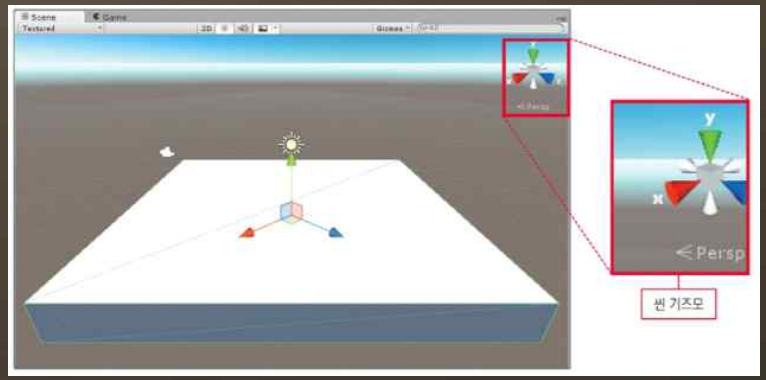
- **이동/회전/확대/축소** – Inspector패널을 이용하여 수치를 직접 입력하면 미세 조정 가능



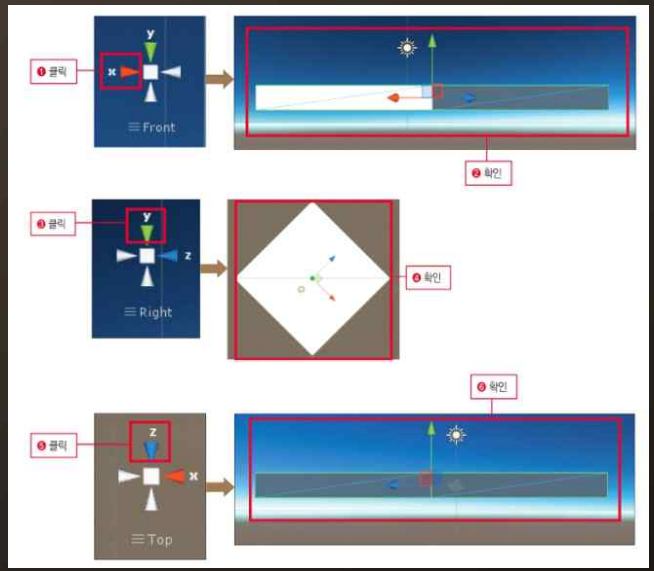
오브젝트 또는 Hierarchy 창에 있는 카메라 등의 위치, 각도, 크기 등의 속성 변환

저작도구 - Unity 3D

- **씬기즈모** – 현재 씬을 촬영하는 카메라가 씬으 어떤 시점으로 보고 있는지 알 수 있음
  (Alt키를 누른 상태에서 오브젝트를 드래그하면 씬기즈모의 모양도 변화)

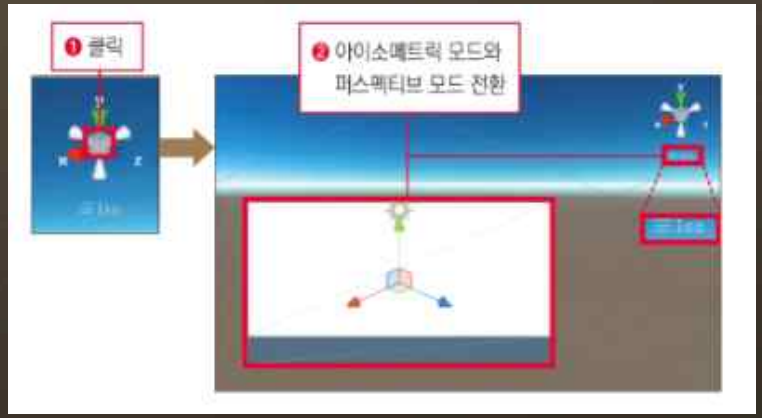


- 시점 바꾸기

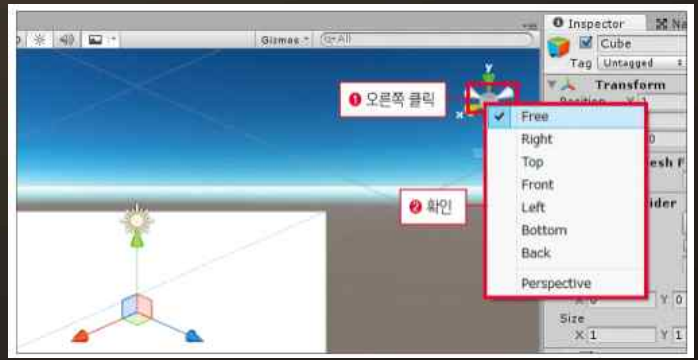


저작도구 - Unity 3D

- Isometric mode & Perspective mode
  - Isometric mode : 멀리 있는 오브젝트 원래 크기 표시
  - Perspective mode : 멀리 있는 것은 작게 가까이에 있는 것은 크게 표시



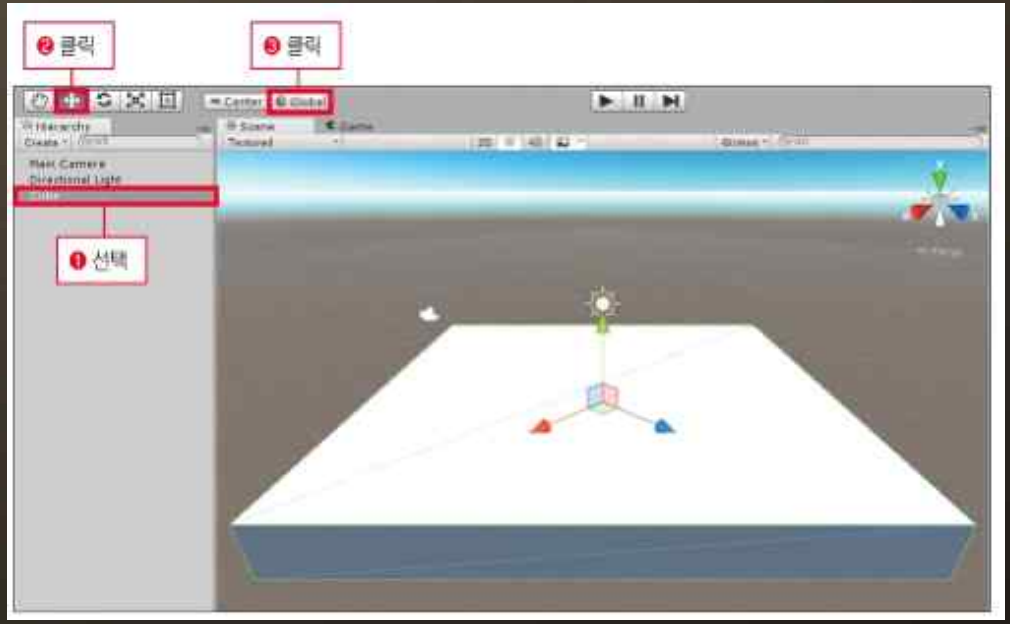
- 카메라 방향 – 마우스 오른 클릭



저작도구 - Unity 3D

- Global Point & Local Point(전역좌표 & 지역좌표)

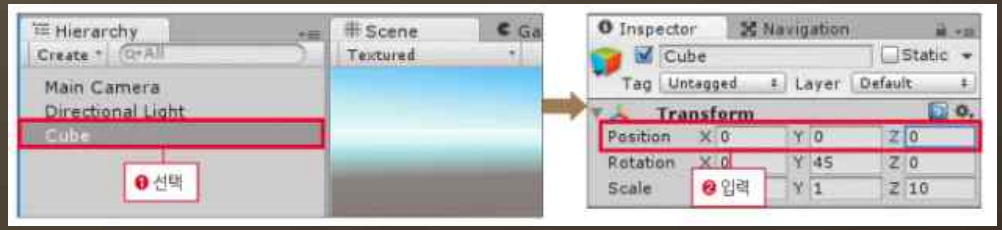
  Hierarchy에서 Cube를 선택하고 이동 도구에서 이동모드를 클릭하여 토글 스위치를 Global과 Local로 전환



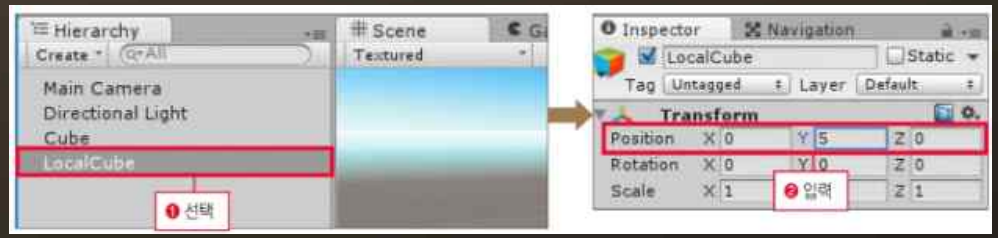
저작도구 - Unity 3D

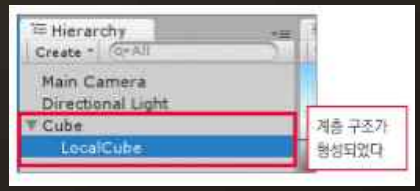
- **계층 구조의 게임 오브젝트 다루기** – 이동 도구에서 계층 구조의 오브젝트를 조작할 때 어느 곳을 중심으로 할지 선택

  - 기존 생성되어있는 Cube 를 선택하여 Inspector에서 Position을 0,0,0으로 변경



  - Hierarchy에서 Create를 클릭–> 3D Object –> Cube 선택
  - Cube 이름 변경–>LocalCube / Position 0,5,0으로 변경
  - Hierarchy에서 LocalCube를 드래그하여 Cube위에 놓는다

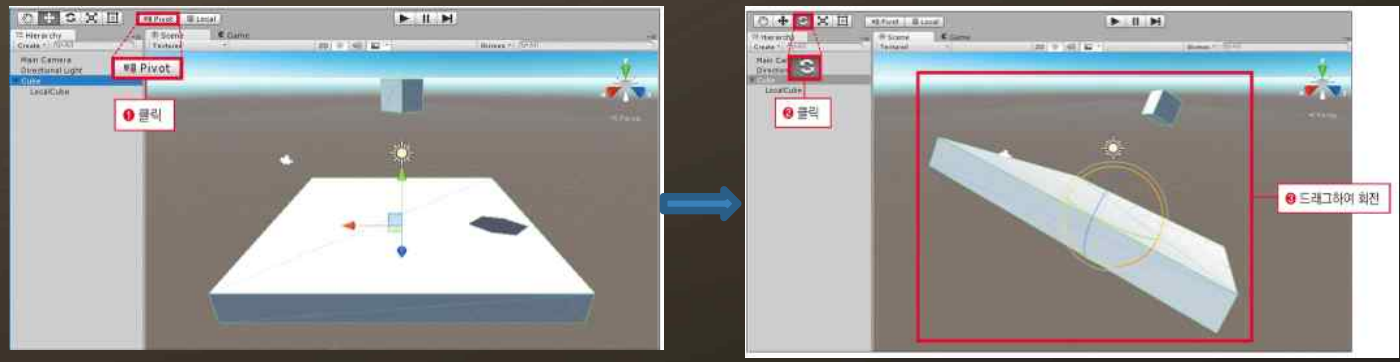




저작도구 - Unity 3D

- **계층 구조의 게임 오브젝트 다루기** – 이동 도구에서 계층 구조의 오브젝트를 조작할 때 어느 곳을 중심으로 할지 선택

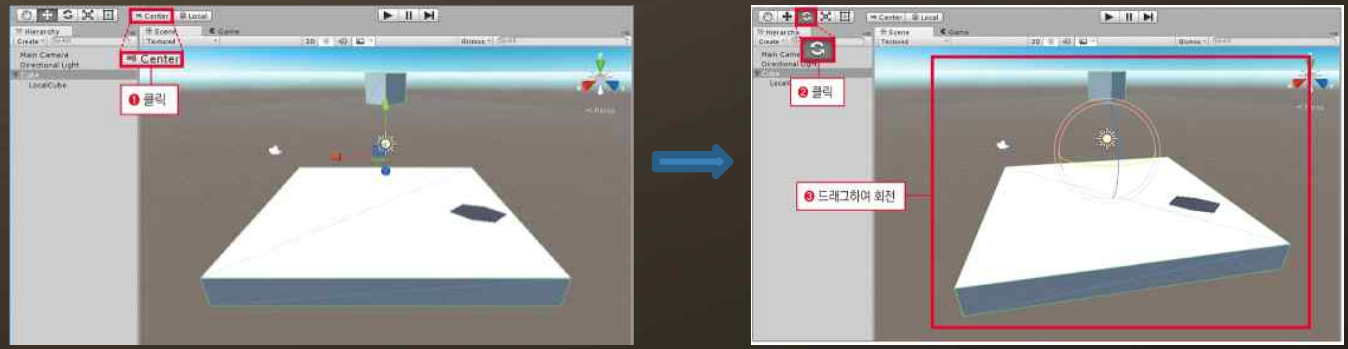
  - 상위 오브젝트인 Cube를 선택하여 토글 스위치를 Pivot과 Center로 각각 전환
  - Pivot을 선택하면 기즈모가 Cube의 중심에 표시.
  - 이동도구를 회전 도구로 전화하고 Alt키를 누른 상태로 드래그 하면 Cube를 중심으로 회전



저작도구 - Unity 3D

- **계층 구조의 게임 오브젝트 다루기** – 이동 도구에서 계층 구조의 오브젝트를 조작할 때 어느 곳을 중심으로 할지 선택

  - 상위 오브젝트인 Cube를 선택하여 토글 스위치를 Pivot과 Center로 각각 전환
  - Center를 선택하면 기즈모가 Cube와 Localcube의 중간에 표시.
  - 이동도구를 회전 도구로 전화하고 Alt키를 누른 상태로 드래그 하면 중간점을 중심으로 회전

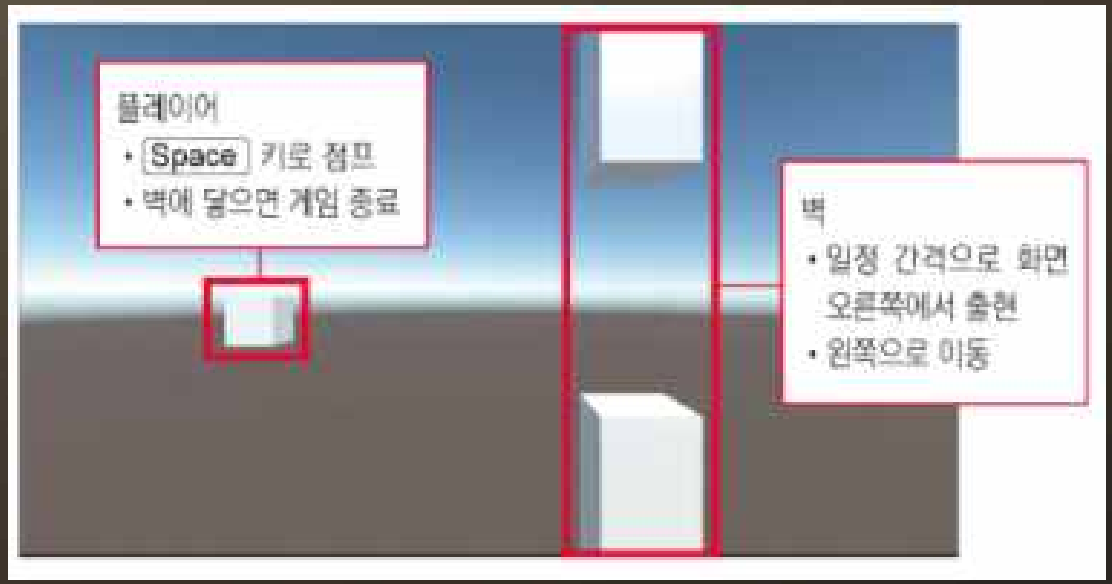


미니 게임 제작

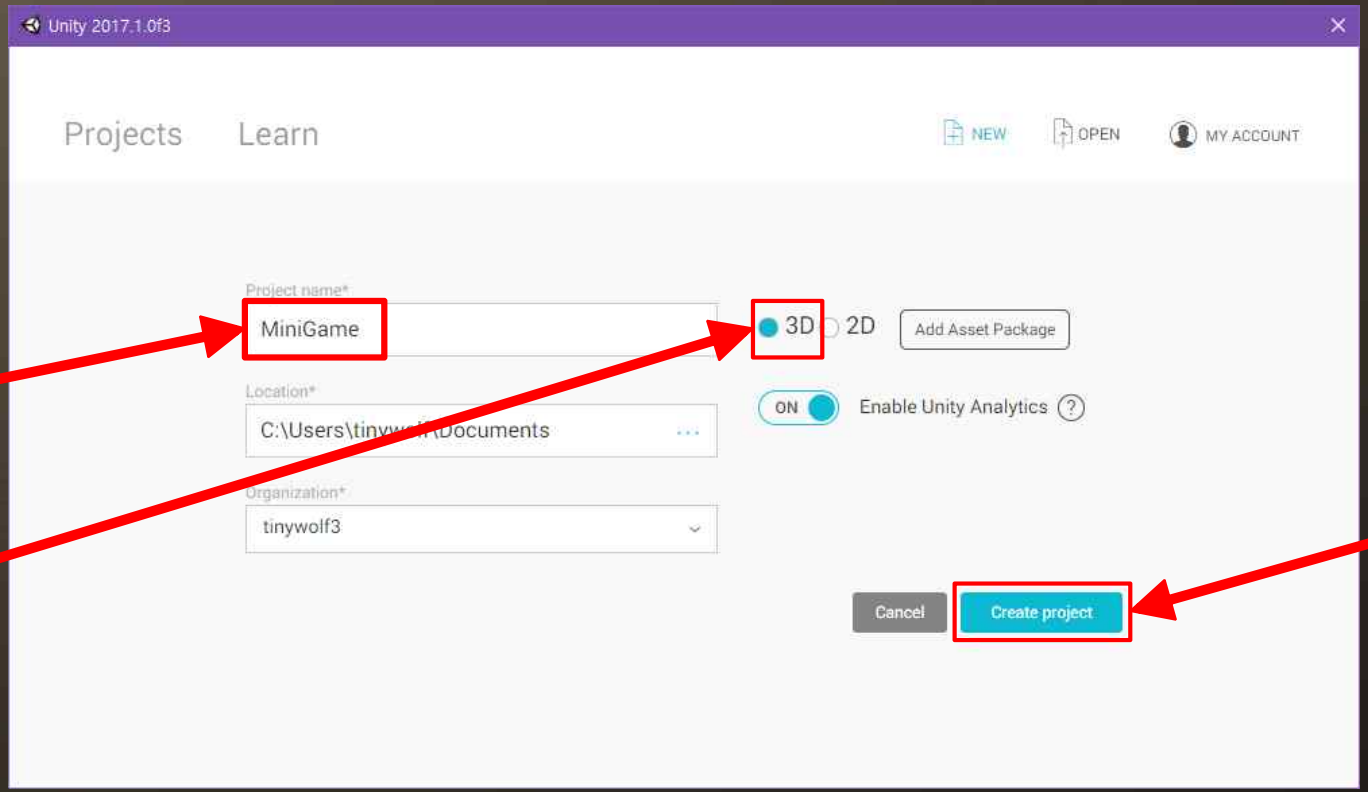
Mini Game Project(with c#)

미니게임 제작

. 벽의 틈새를 점프로 뛰며 전진하는 미니게임

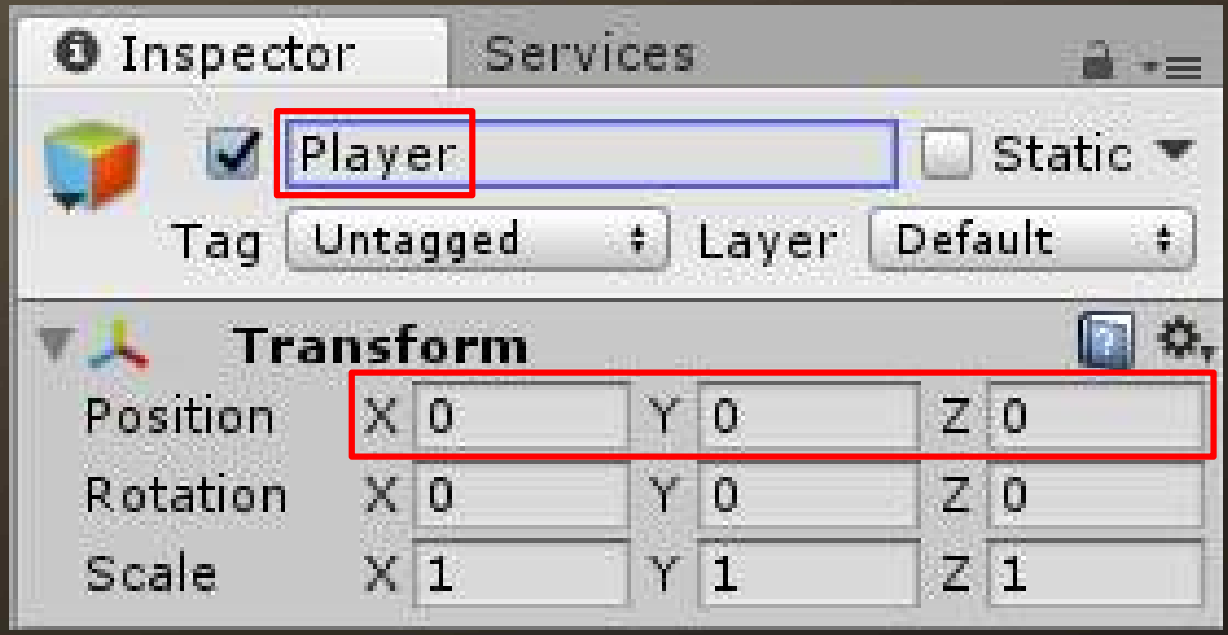


프로젝트 생성

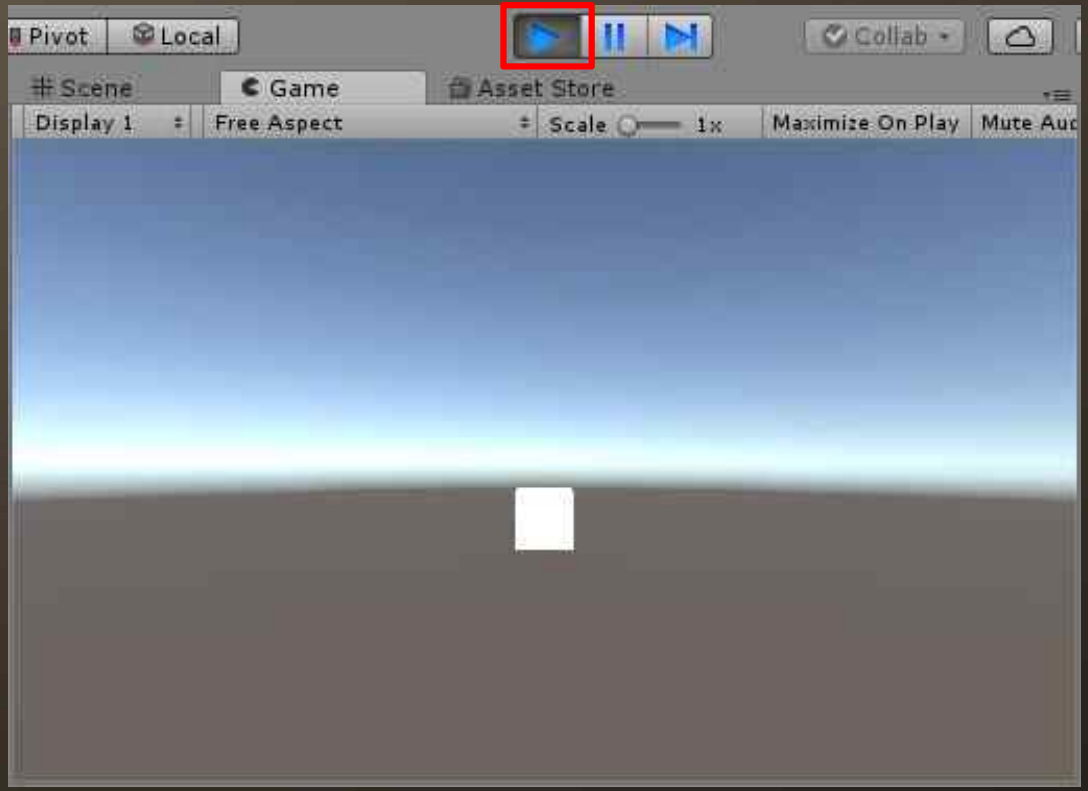

Unity 2017.1.0f3
Projects
Learn
NEW
OPEN
MY ACCOUNT
Project name*
MiniGame
3D
2D
Add Asset Package
Location*
C:\Users\tinyw
Documents
ON
Enable Unity Analytics
Organization*
tinywolf3
Cancel
Create project
1
2
3

오브젝트 추가

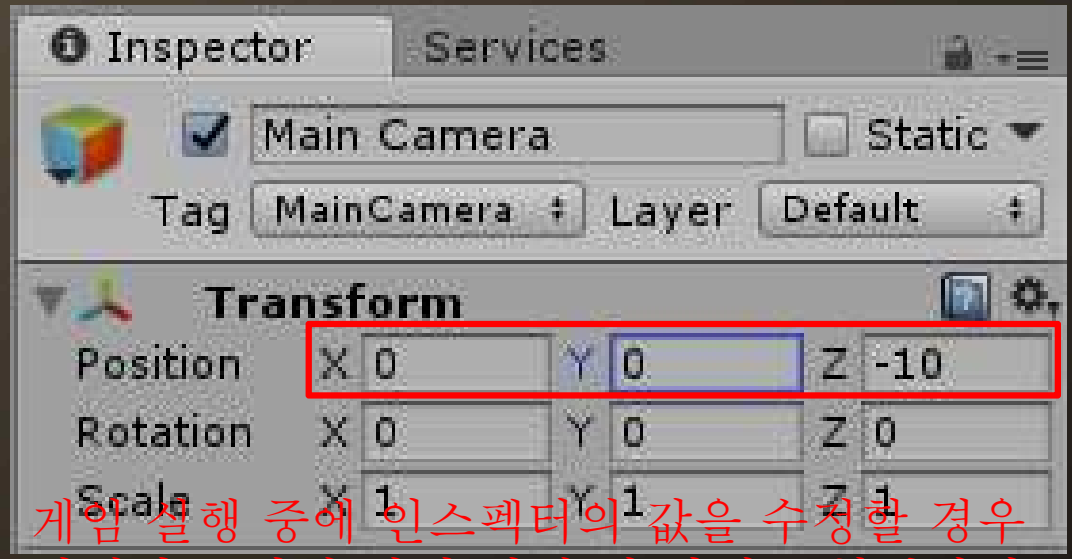
1. GameObject > 3D Object > Cube



플레이

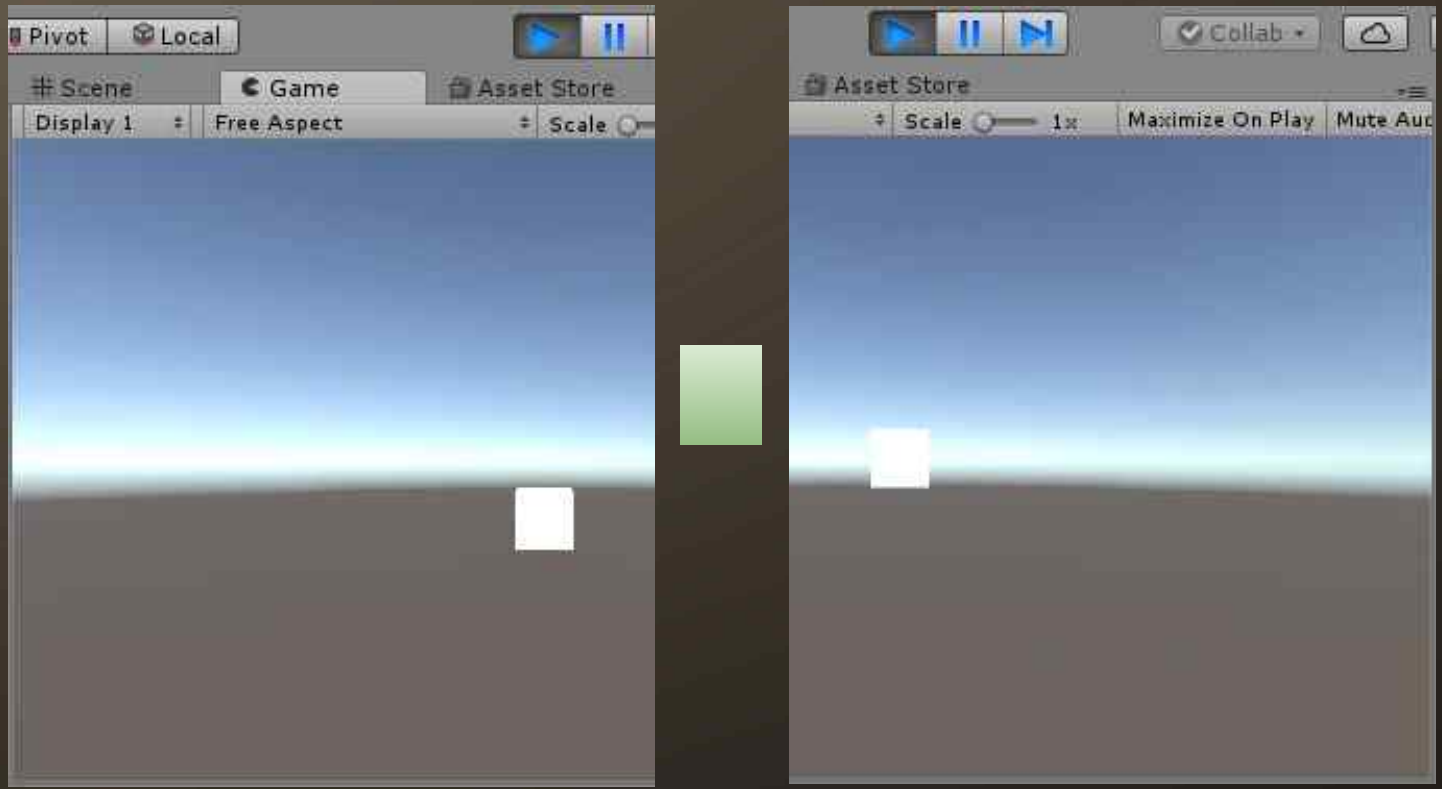

Pivot
Local
Collab
Scene
Game
Asset Store
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Auc

메인 카메라 위치 조정



게임 실행 중에 인스펙터의 값을 수정할 경우
실행이 끝나면 값이 실행 전 상태로 복귀하기 때문에
완전히 수정하려면 게임 실행을 중단하고 수

플레이

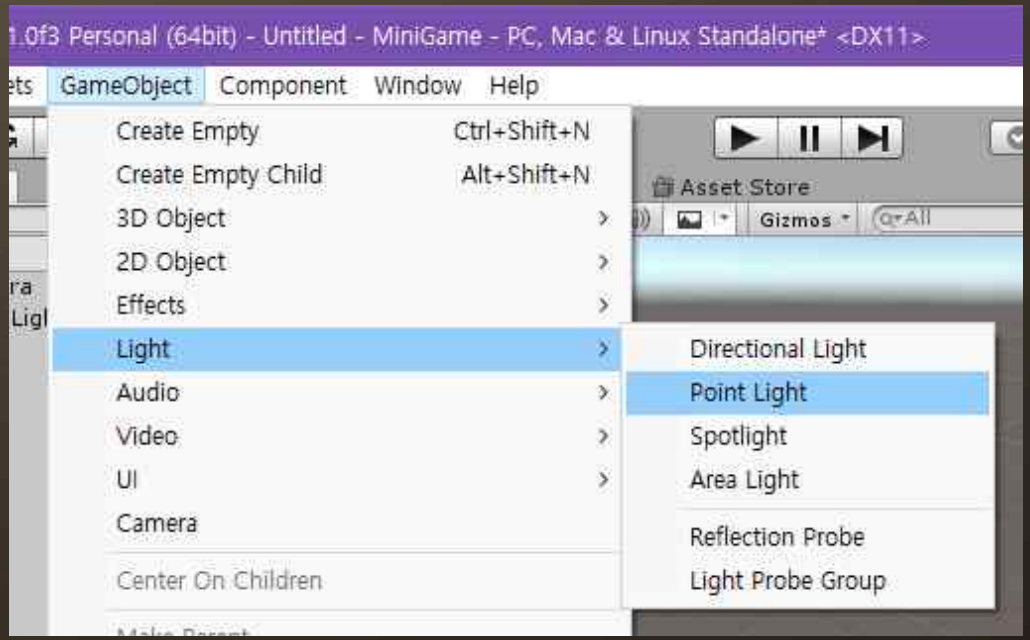

Pivot
Local
Scene
Game
Asset Store
Display 1
Free Aspect
Scale
Collab
Asset Store
Scale
1x
Maximize On Play
Mute Aud

조명

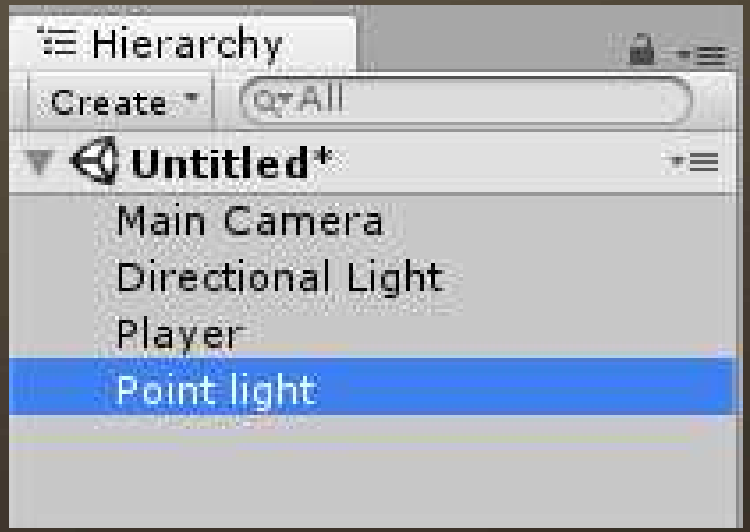
Lights

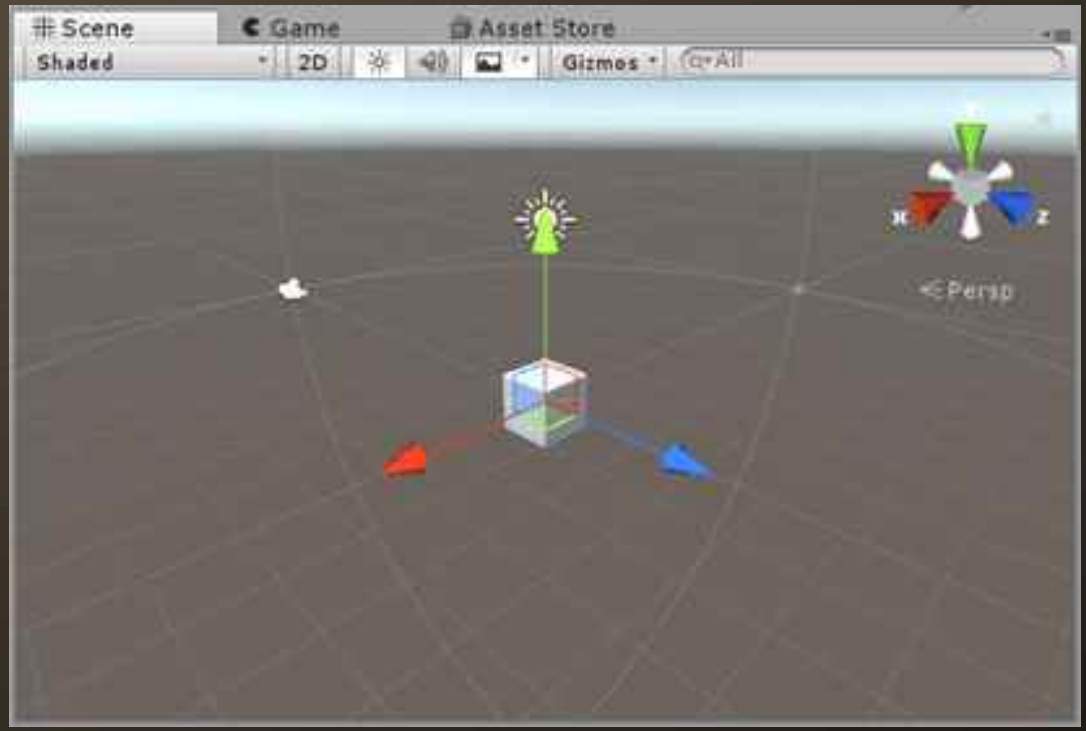
조명 추가

2. GameObject > Light > Point Light

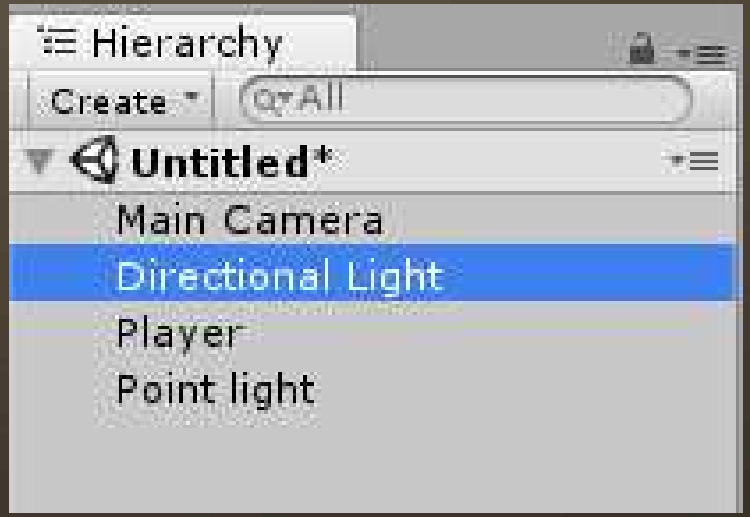


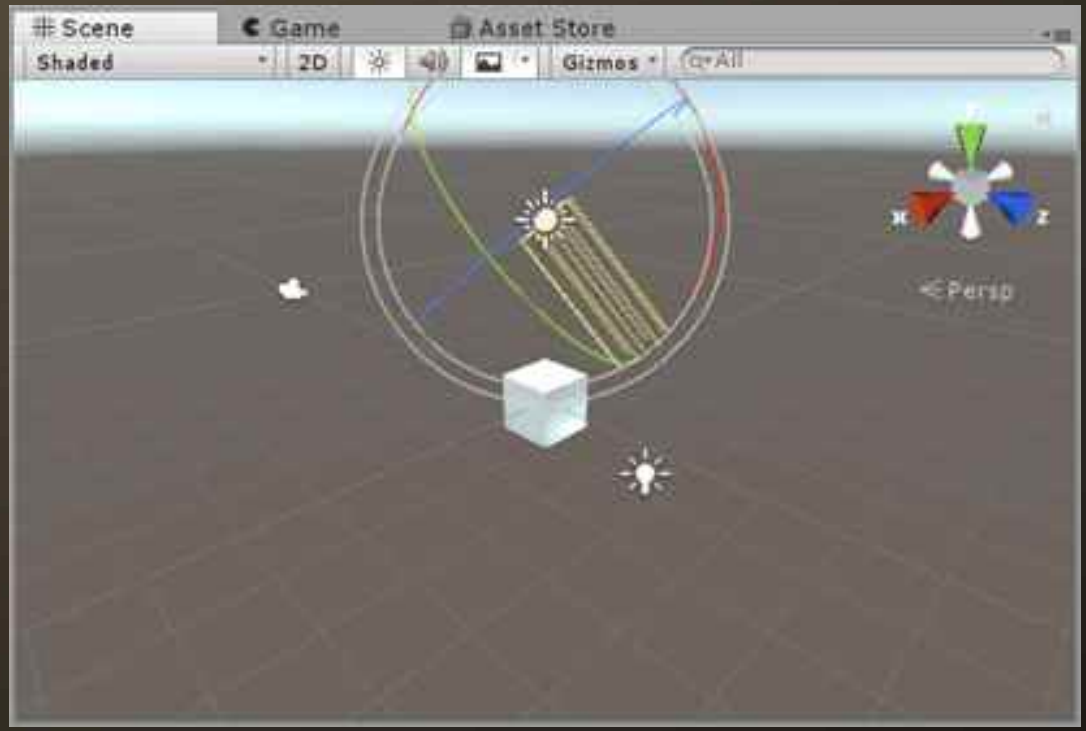
조명 위치 조정


Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
Player
Point light


Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Persp

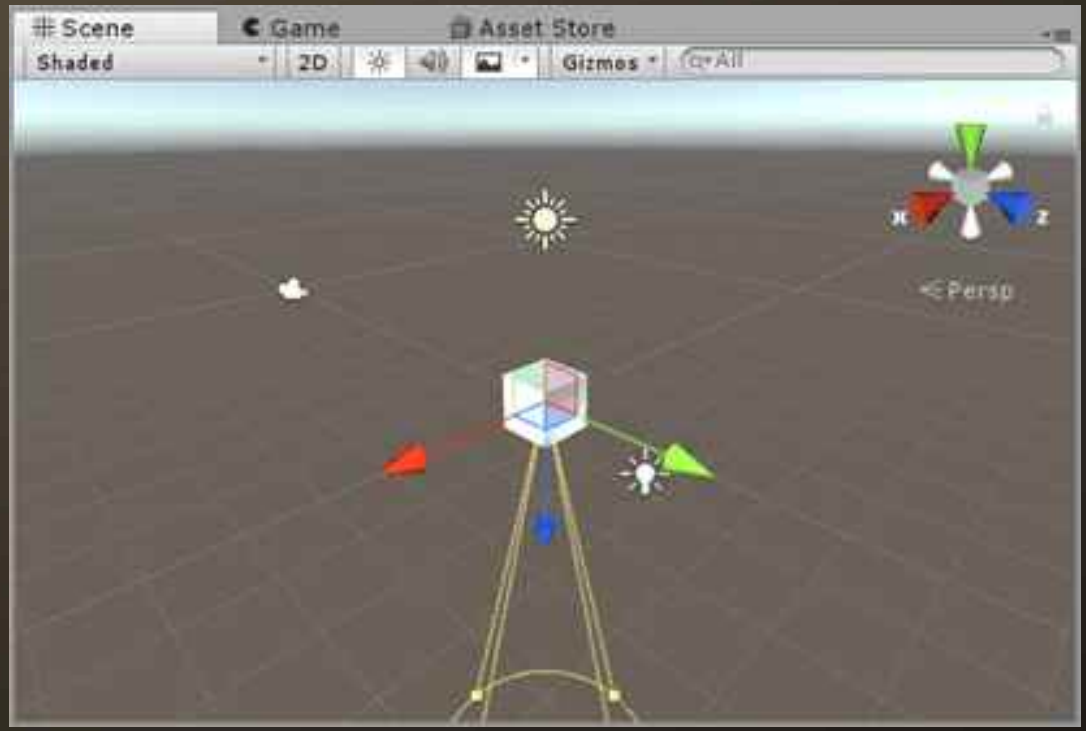
조명 위치 조정


Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
Player
Point light

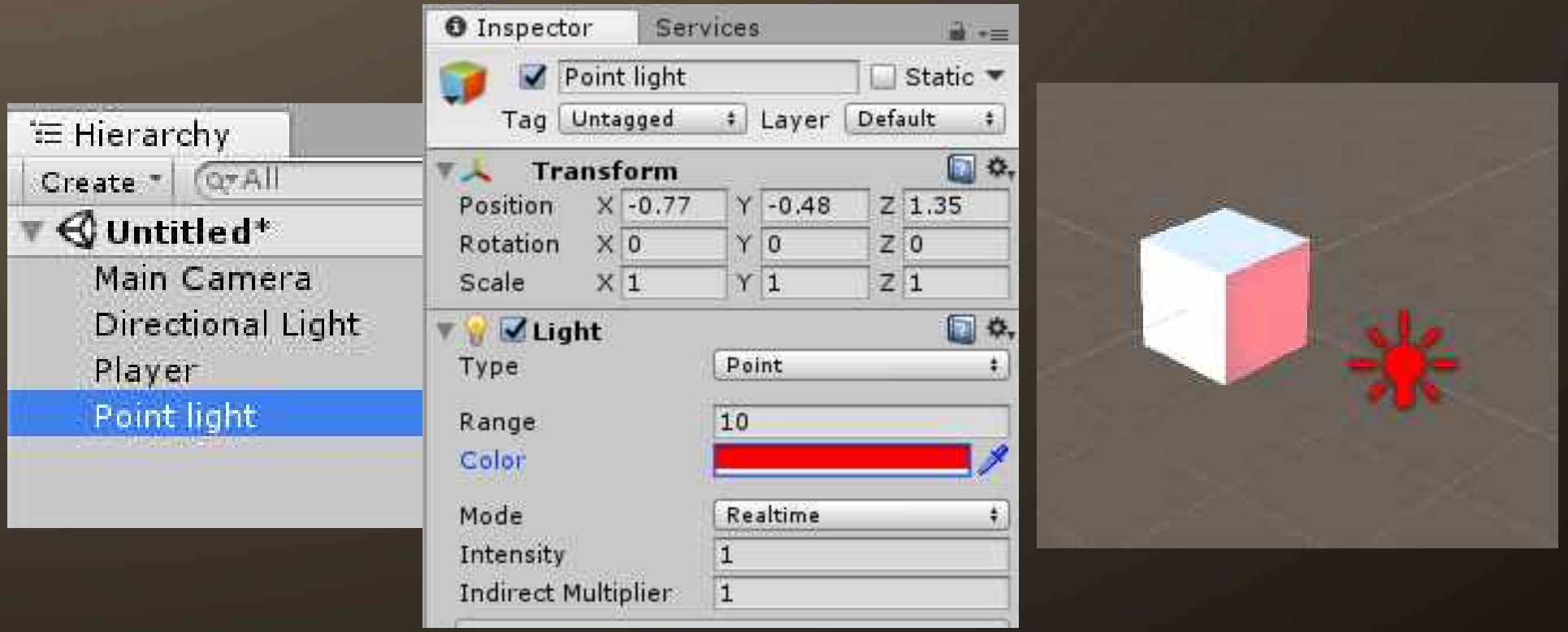

Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Persp

조명 위치 조정

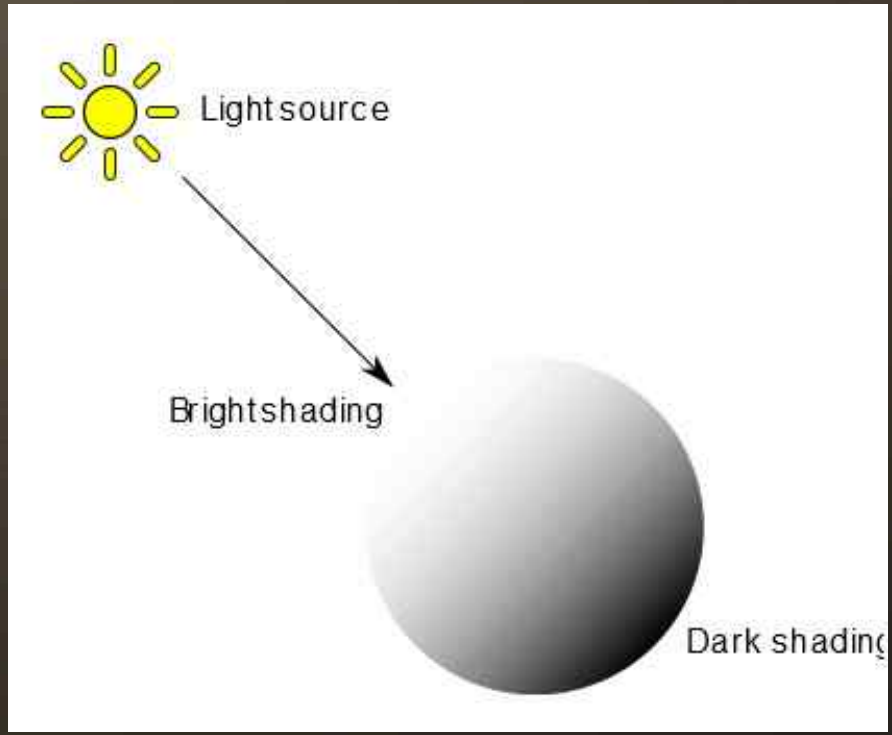

Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
Player
Point light
Spotlight


Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Persp

조명에 색상 넣기

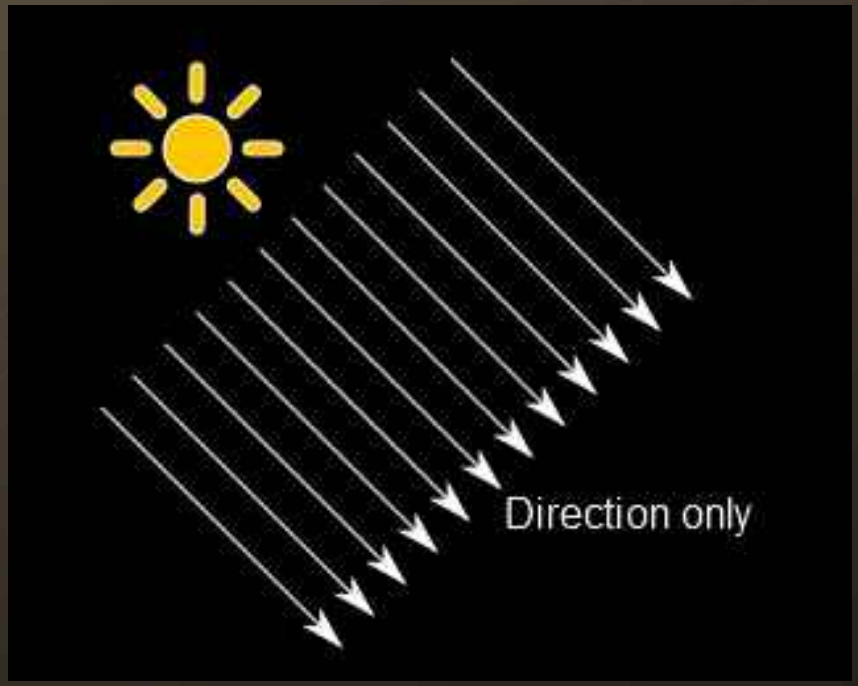

Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
Player
Point light
Inspector
Services
Point light
Static
Tag
Untagged
Layer
Default
Transform
Position
X -0.77
Y -0.48
Z 1.35
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Light
Type
Point
Range
10
Color
Mode
Realtime
Intensity
1
Indirect Multiplier
1

조명(Light)이란

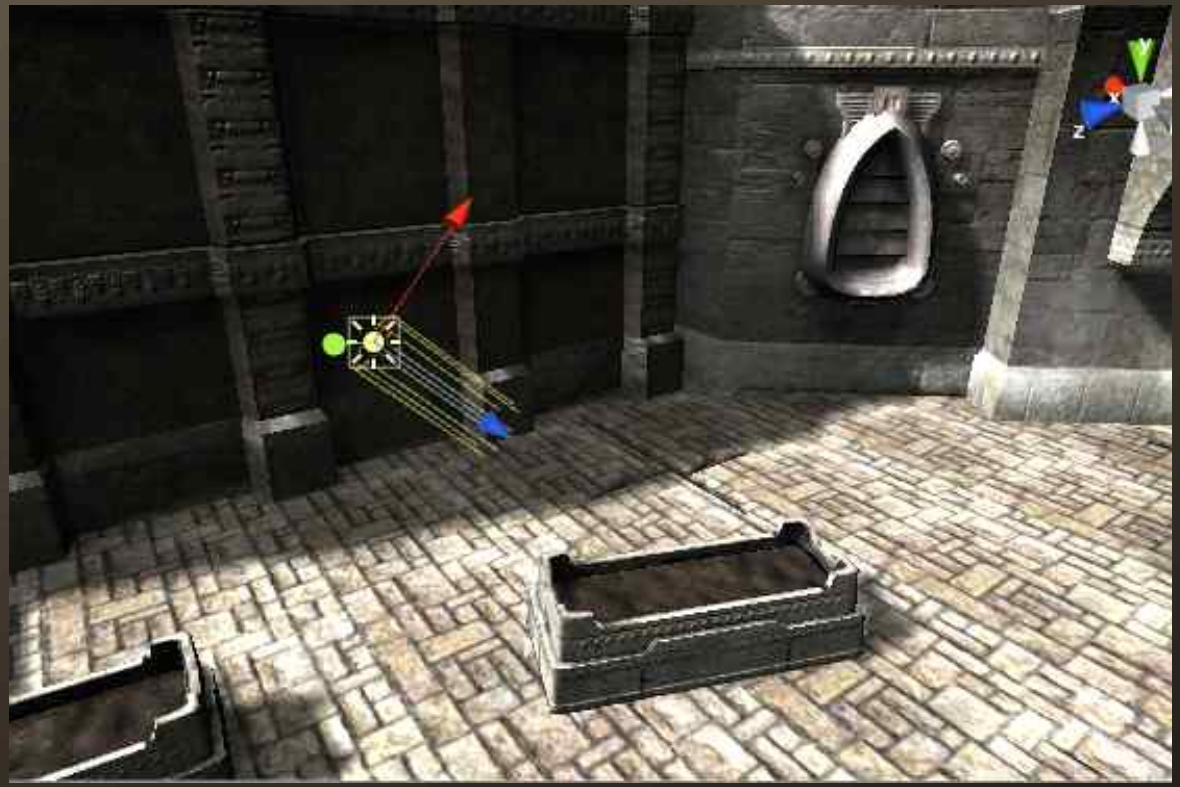


조명의 종류

- Directional Light

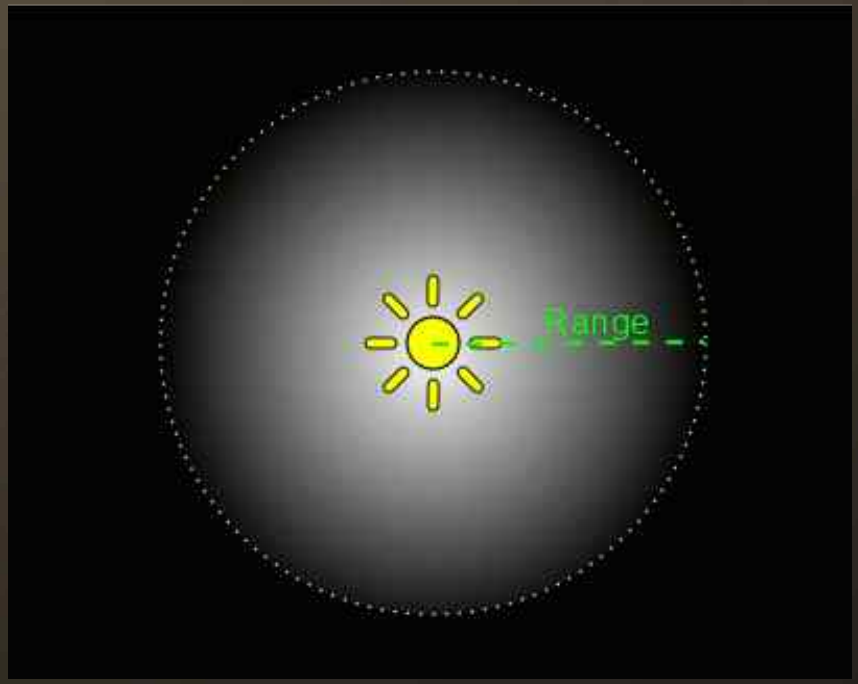


조명의 종류

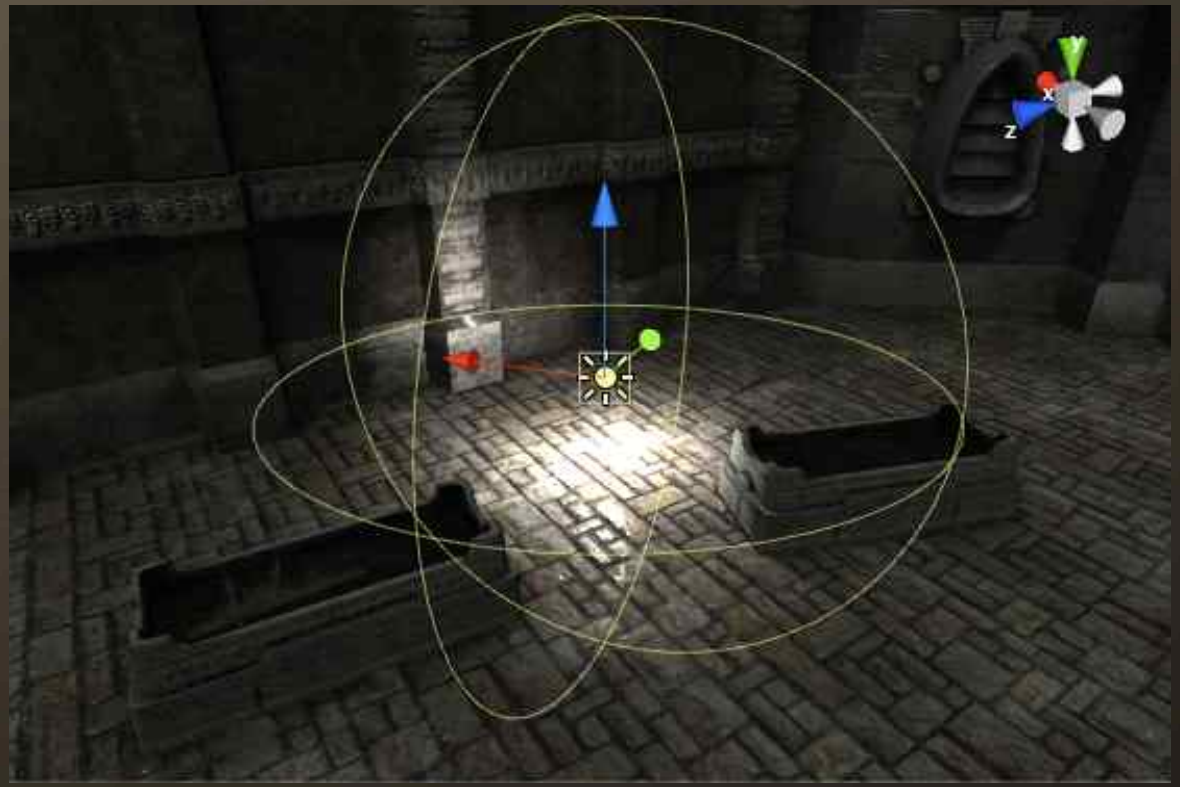


조명의 종류

- Point Light

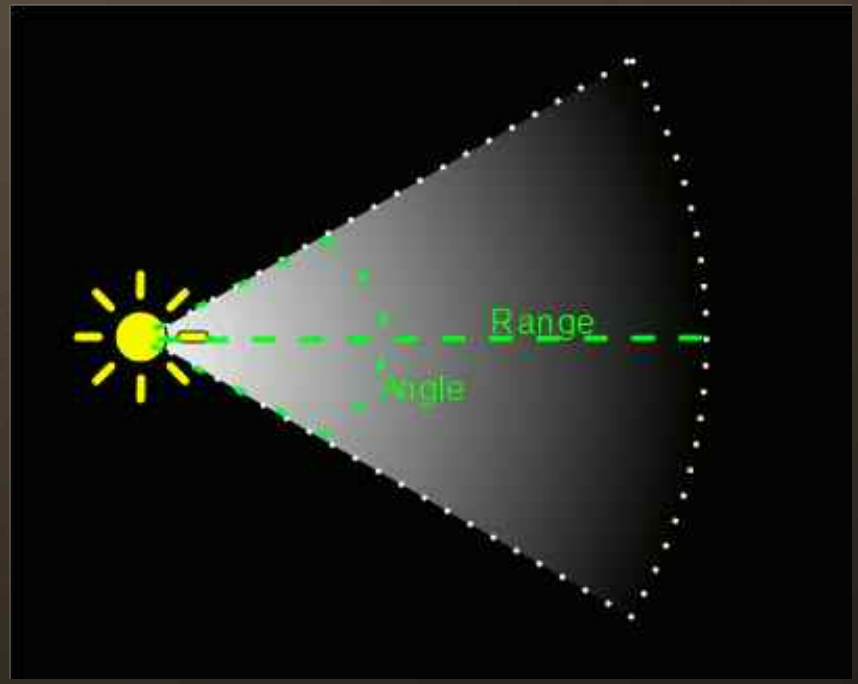


조명의 종류

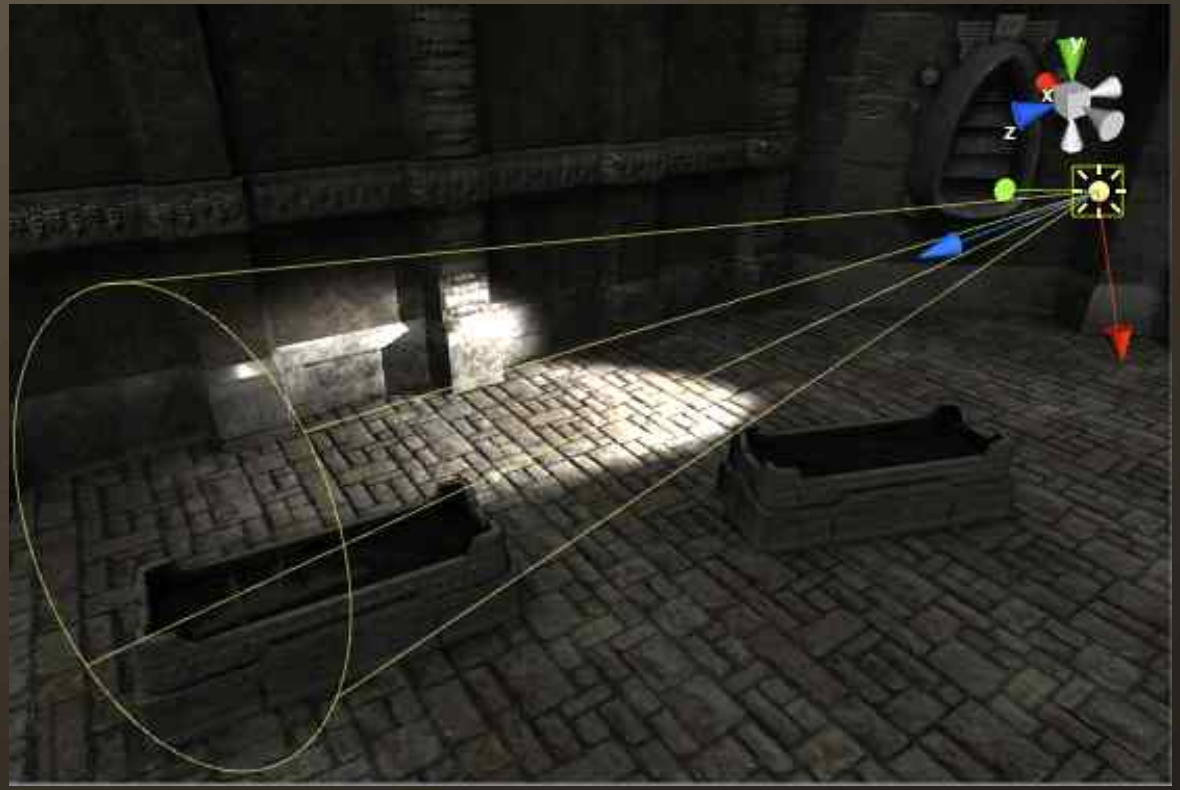


조명의 종류

- Spot Light

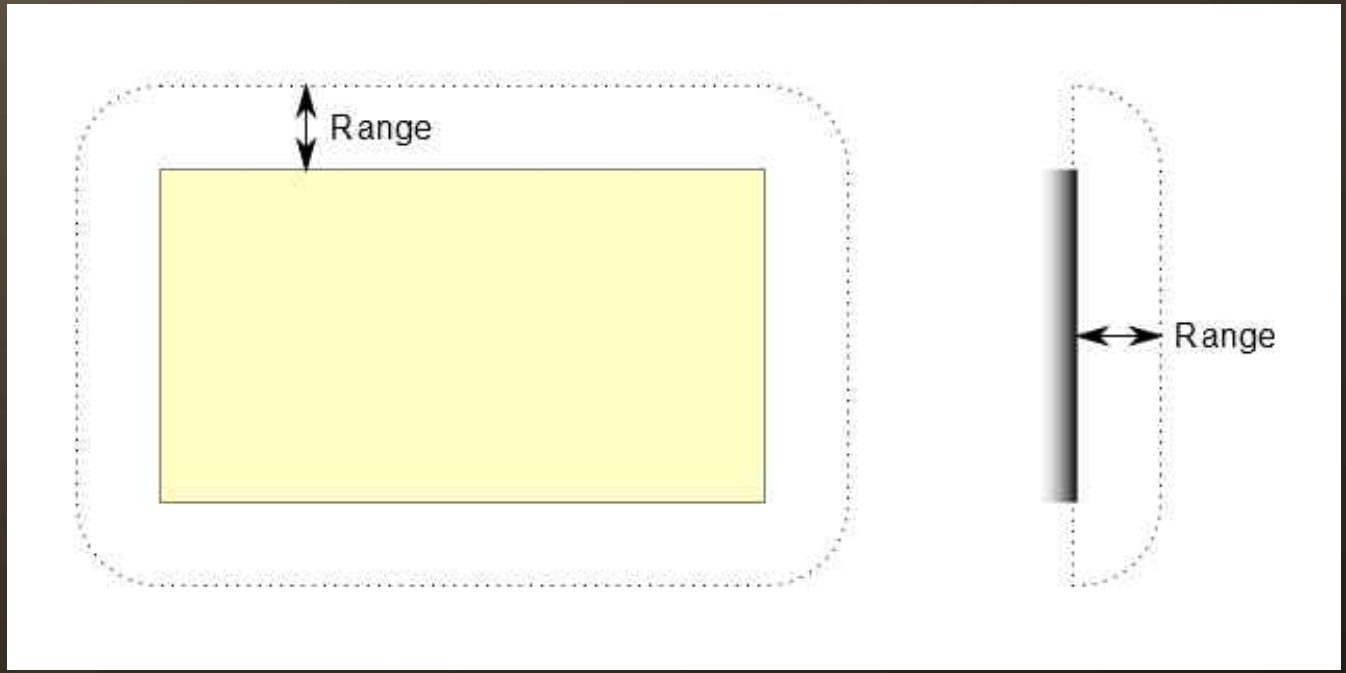


조명의 종류



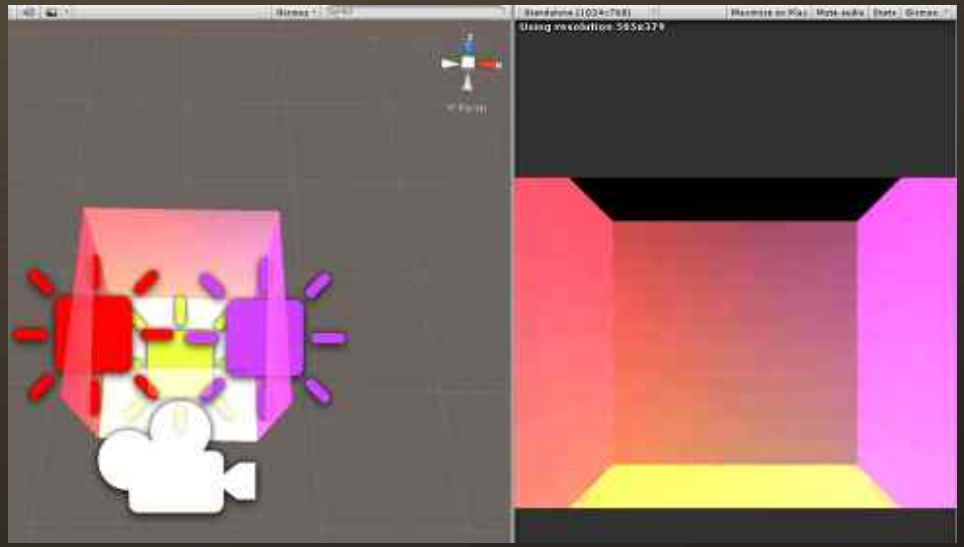
조명의 종류

- Area Light

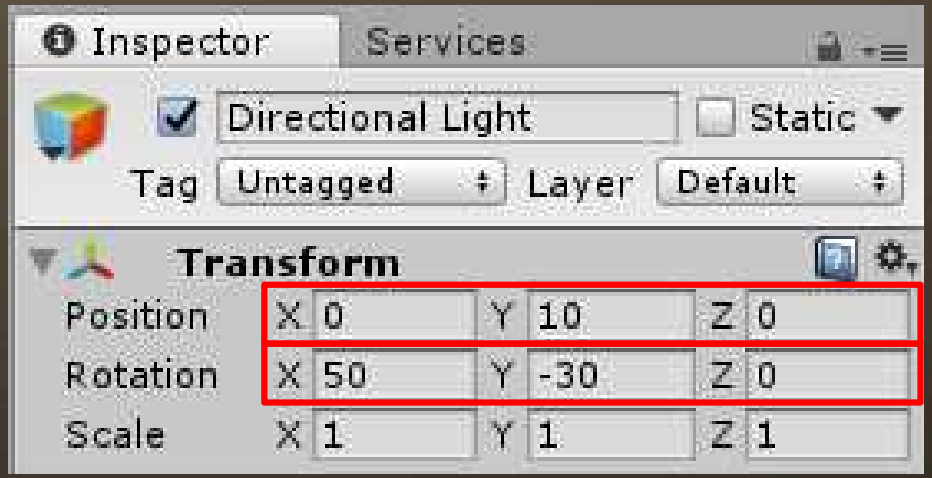


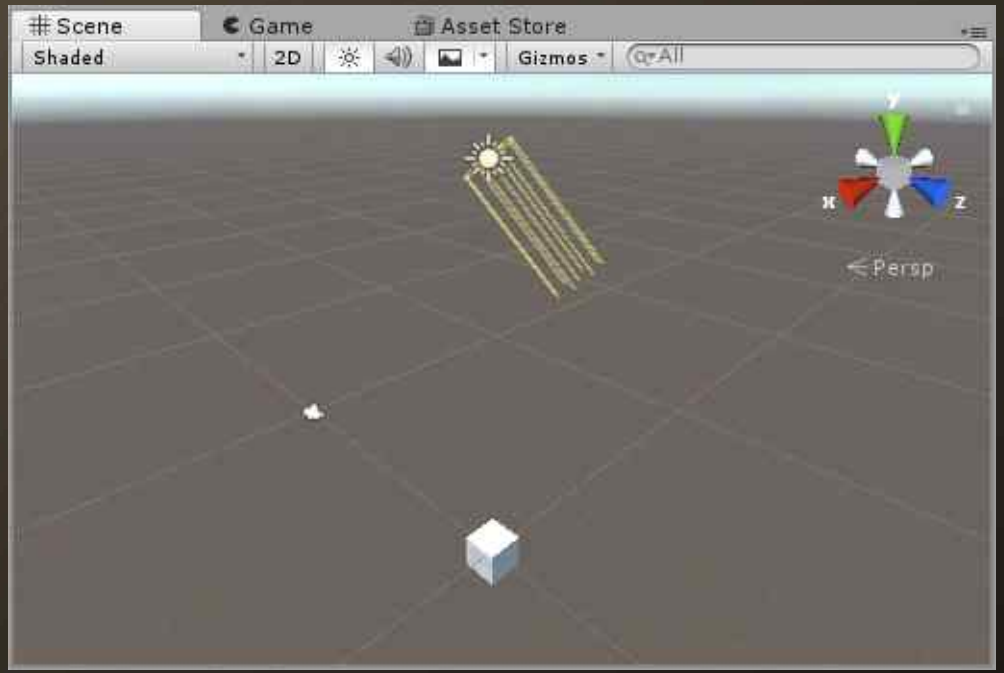
Area Light

- 참고 자료
  - http://dlgnlfus.tistory.com/89

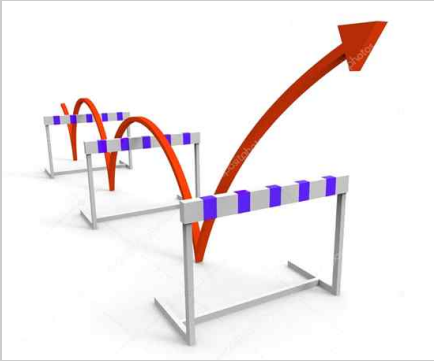


메인 조명 위치


Inspector
Services
Directional Light
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 10
Z 0
Rotation
X 50
Y -30
Z 0
Scale
X 1
Y 1
Z 1


Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp

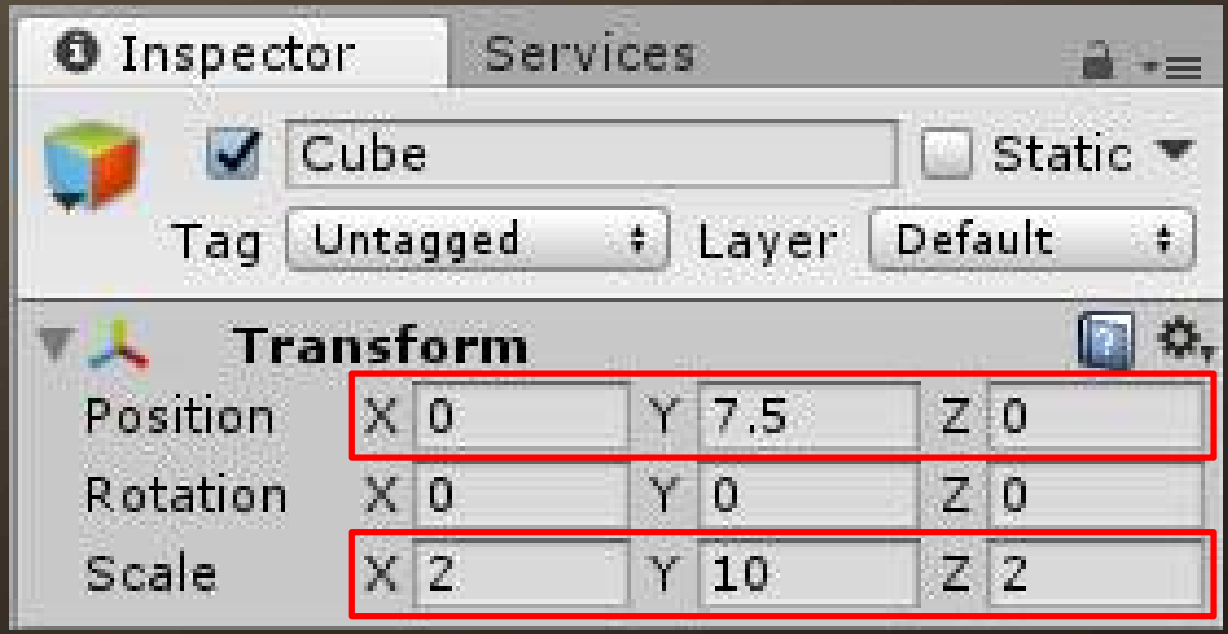
장애물



Obstacles

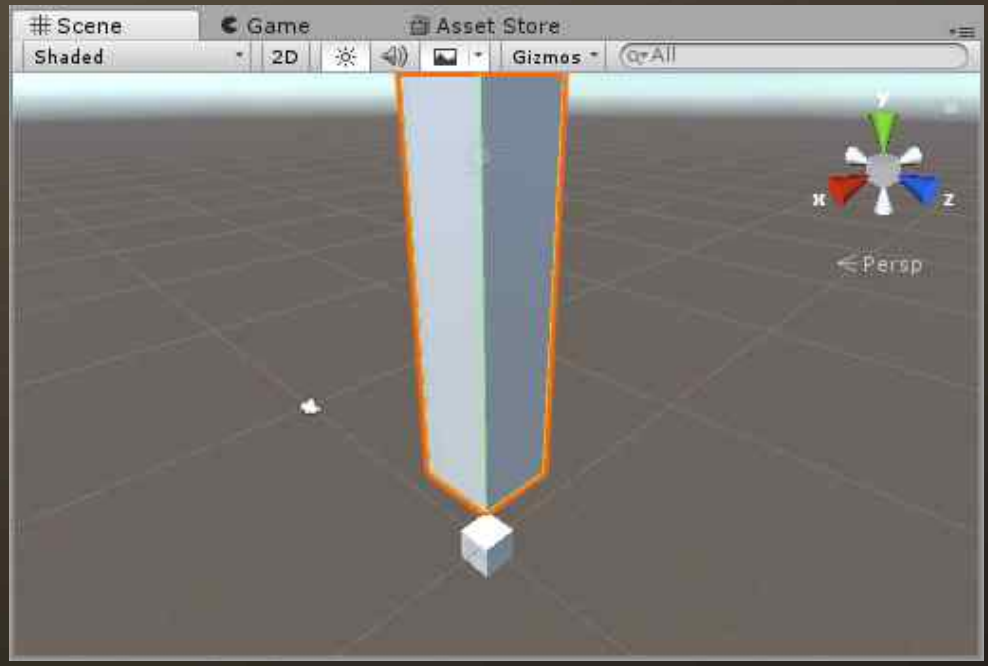
장애물 벽 추가

3. GameObject > 3D Object > Cube



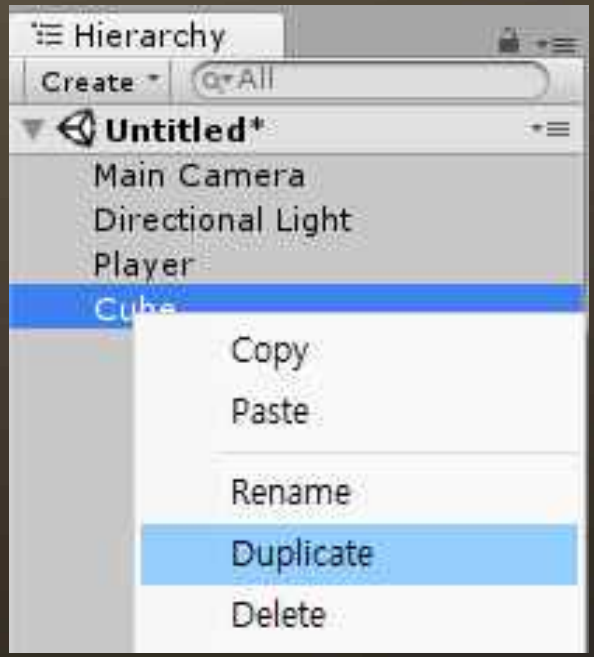
장애물 벽 추가

3. GameObject > 3D Object > Cube



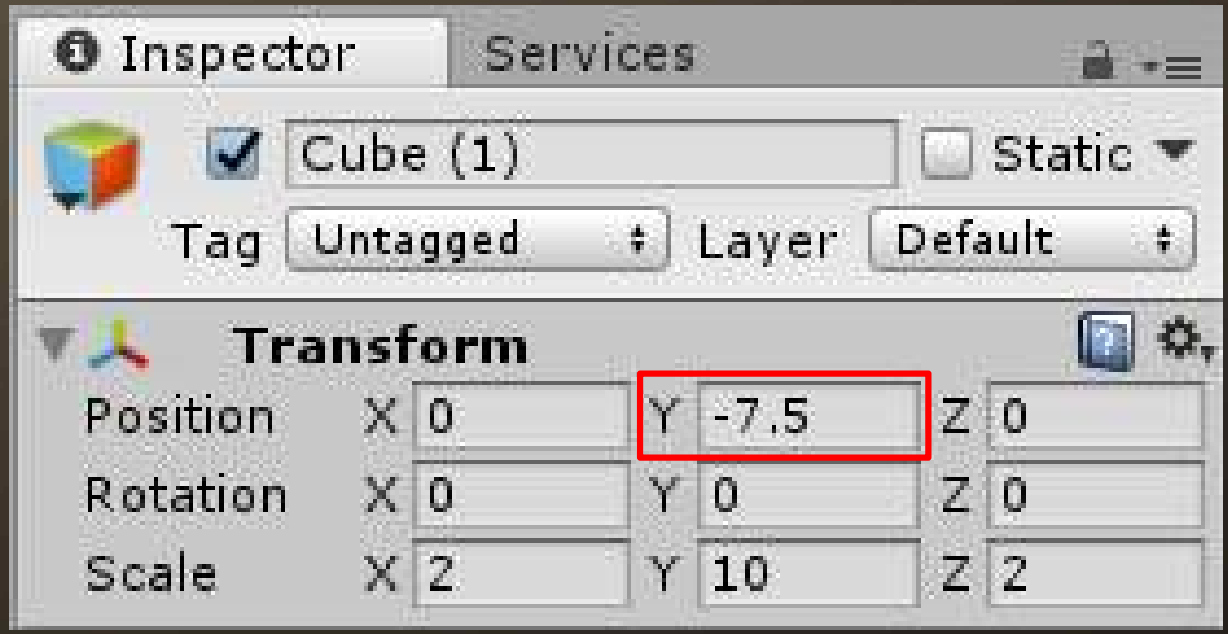
벽 복제

4. Cube > Duplicate



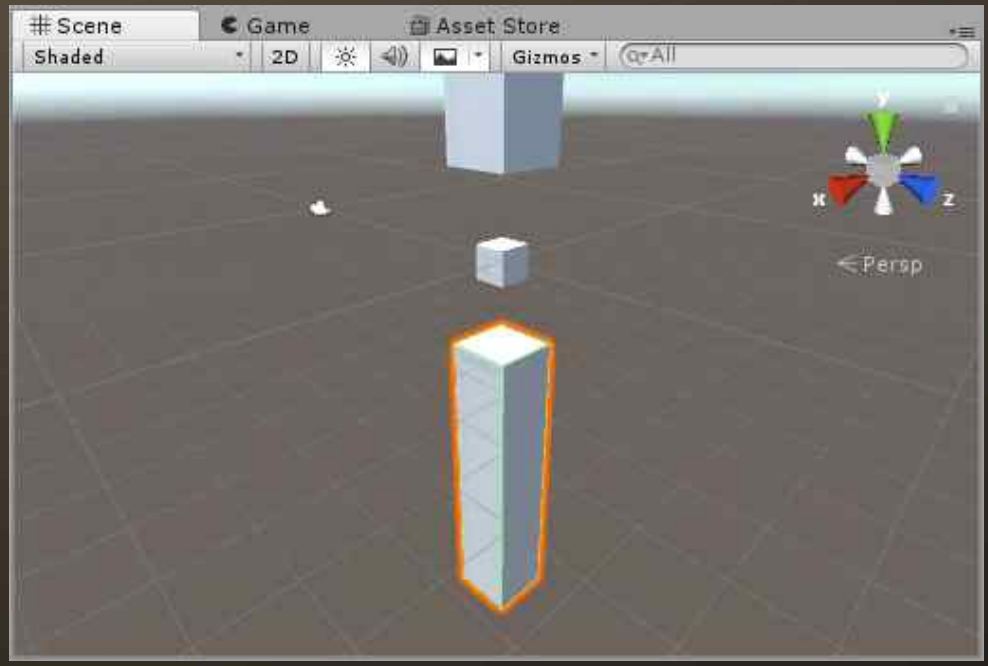
벽 배치

4. Cube > Duplicate



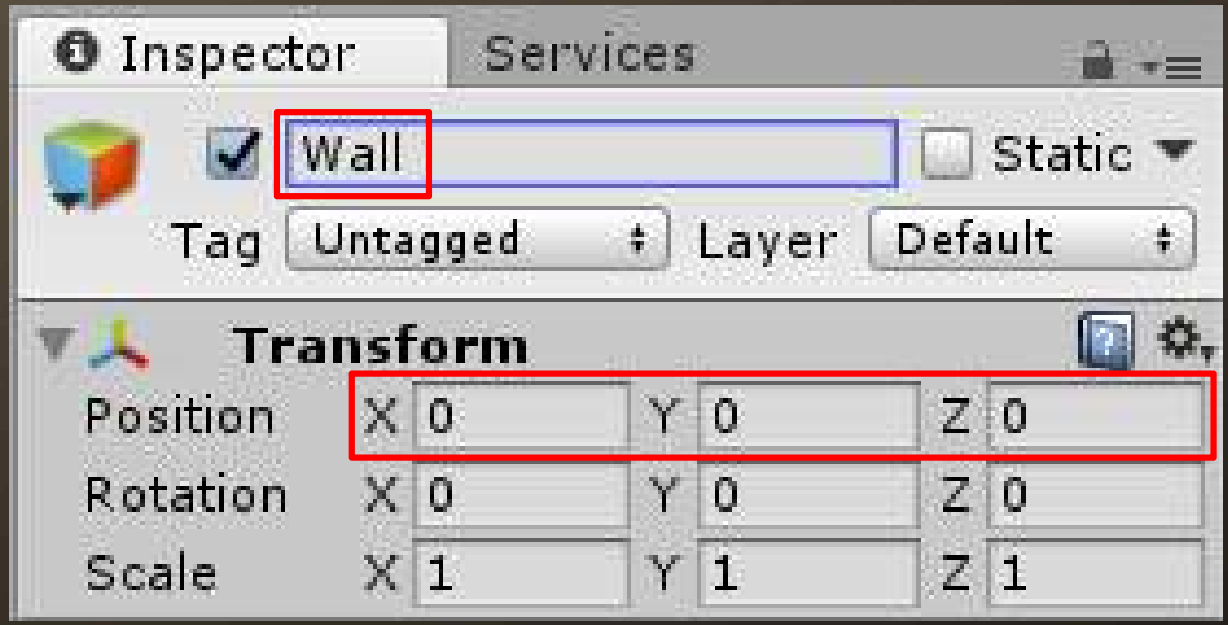
벽 배치

4. Cube > Duplicate



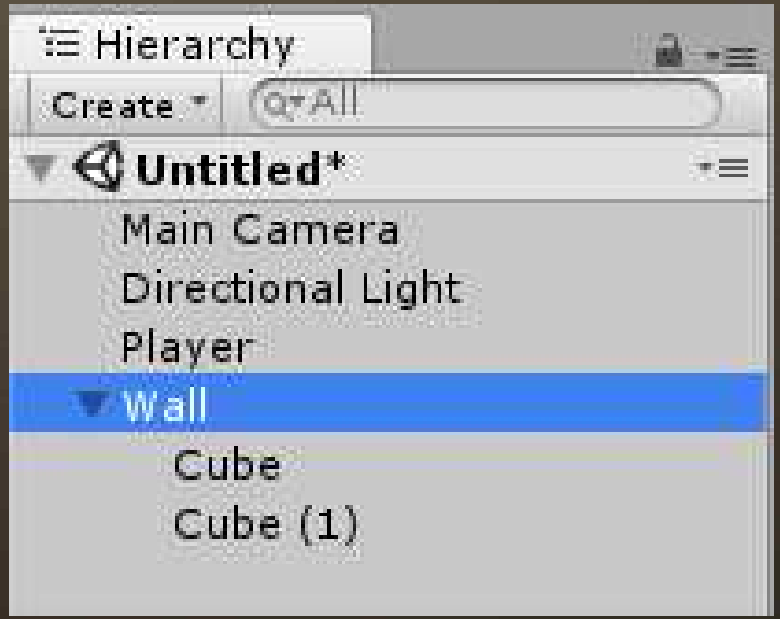
빈 오브젝트 추가

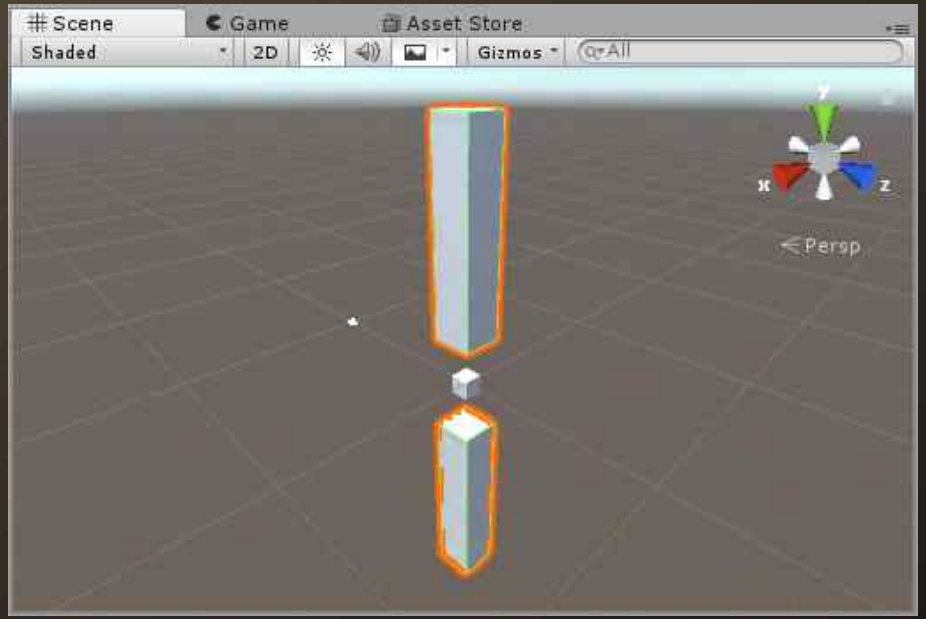
5. GameObject > Create Empty



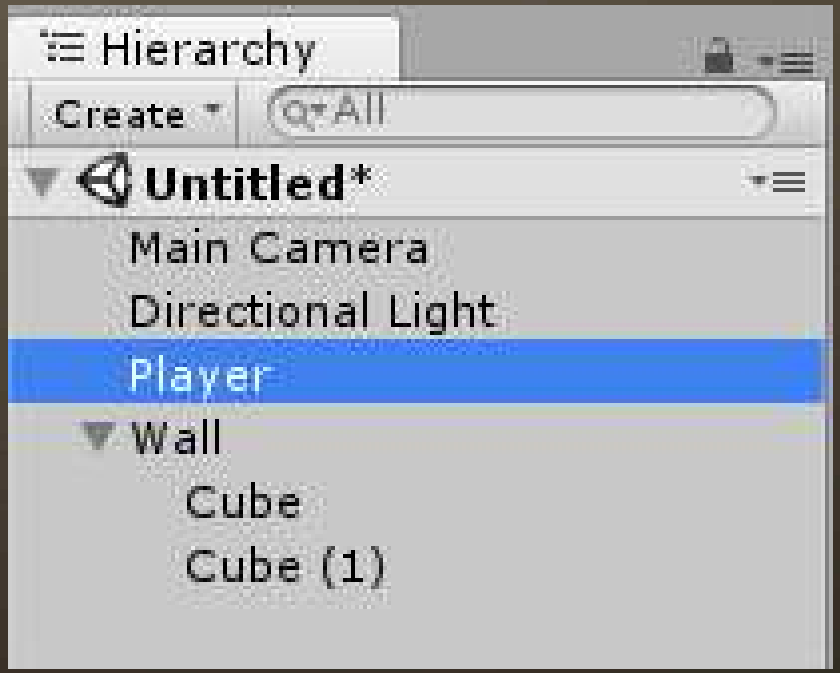
벽 오브젝트 조정

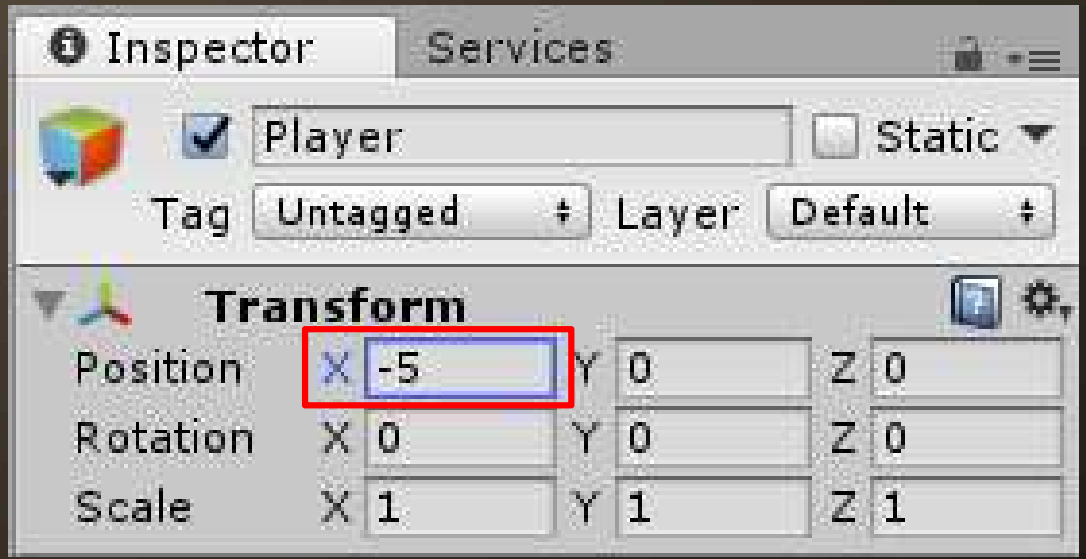
5. GameObject > Create Empty



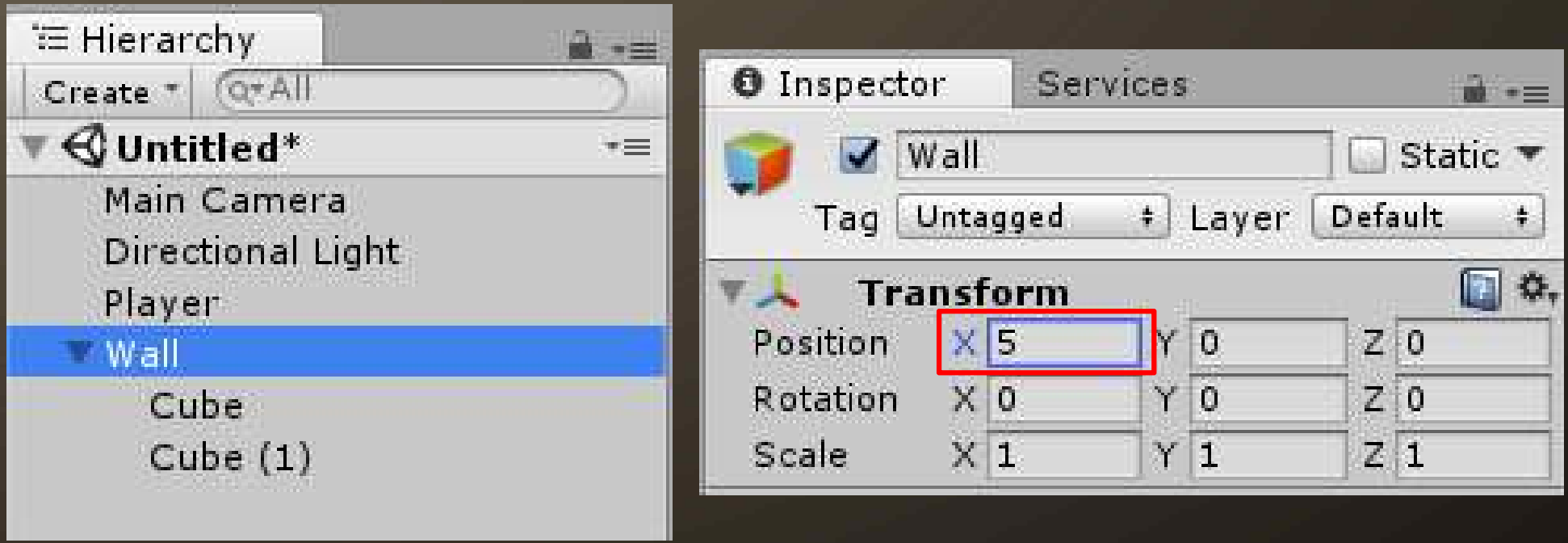


플레이어와 벽의 위치 조정

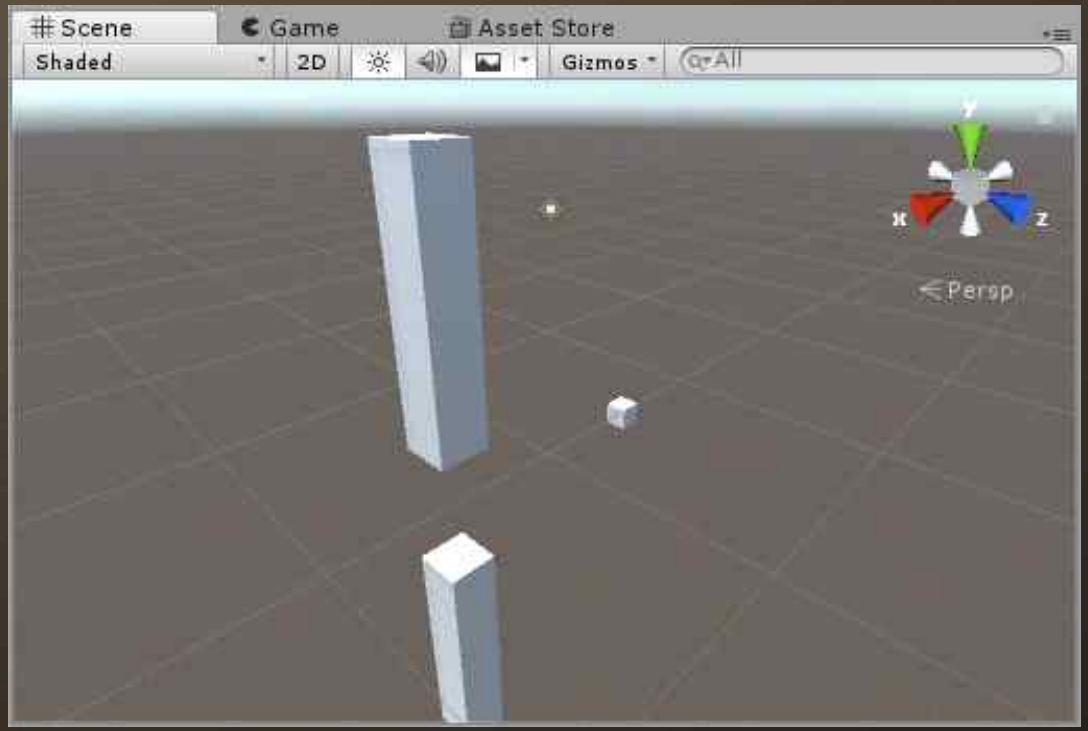




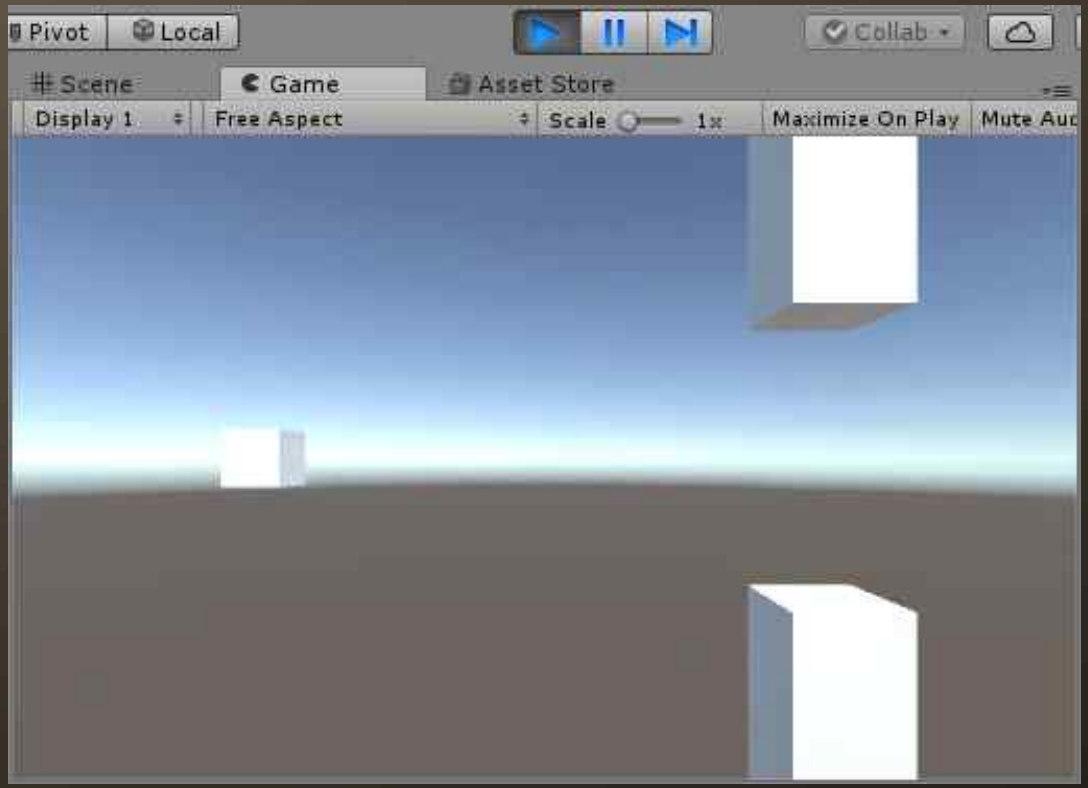
플레이어와 벽의 위치 조정



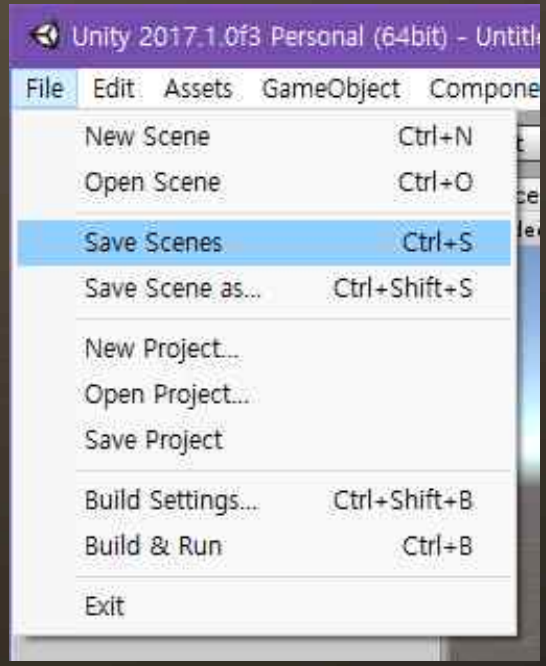
플레이어와 벽의 위치 조정



플레이

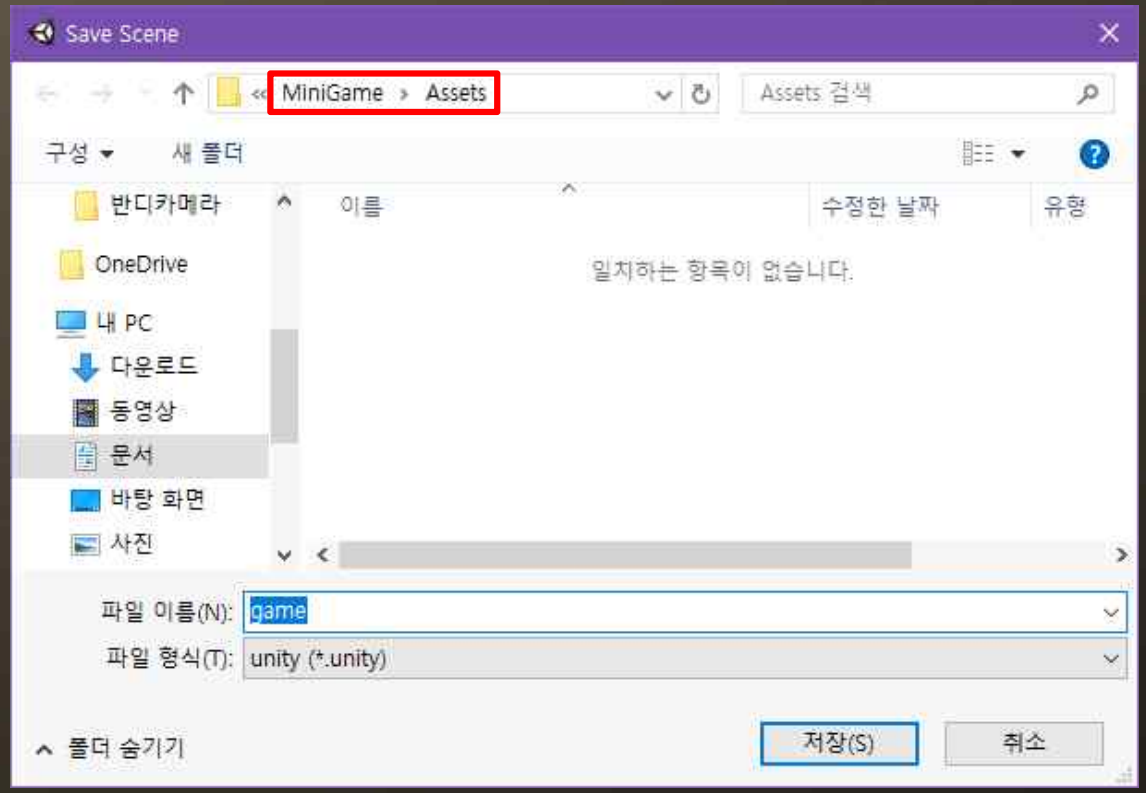

Pivot
Local
Collab
Scene
Game
Asset Store
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Au

장면 저장

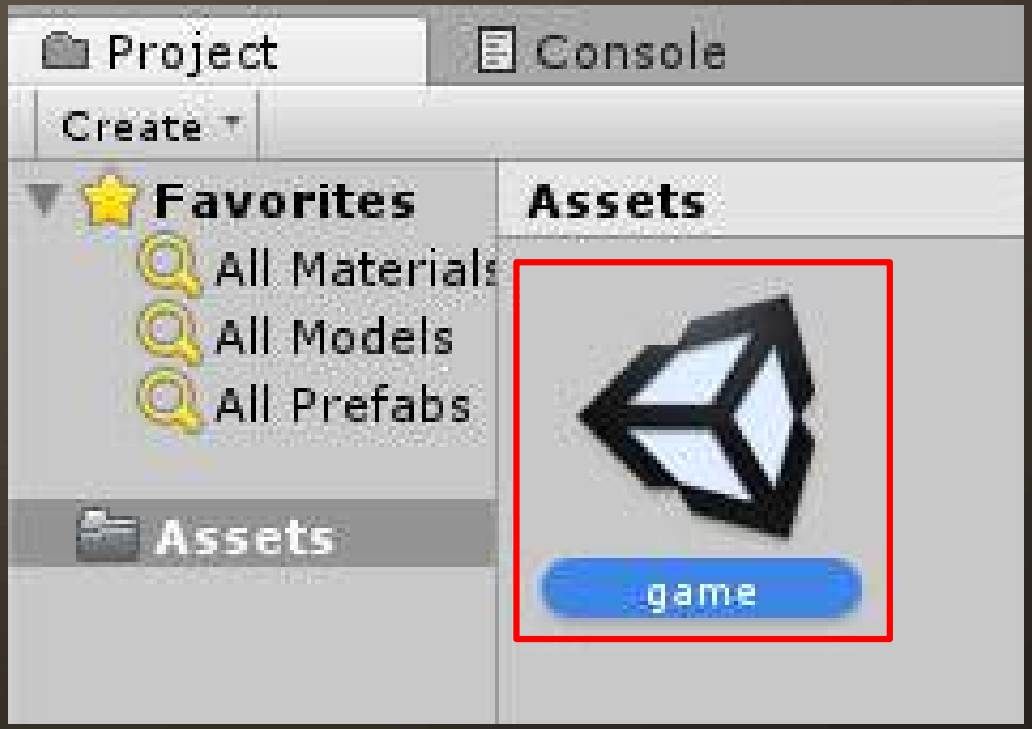


Menu: File > Save Scenes (Ctrl+S)

장면 저장

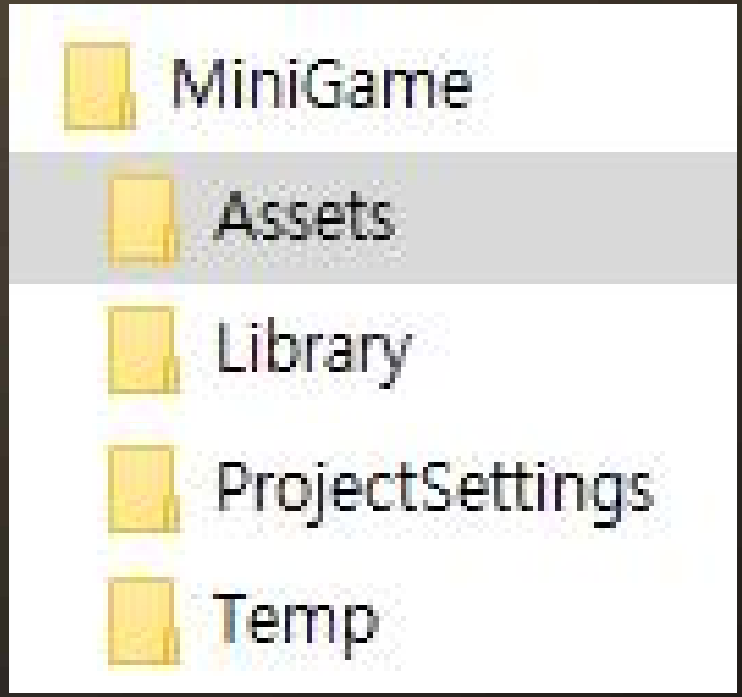

Save Scene
MiniGame › Assets
Assets 검색
구성
새 폴더
반디카메라
OneDrive
내 PC
다운로드
동영상
문서
바탕 화면
사진
이름
수정한 날짜
유형
일치하는 항목이 없습니다.
파일 이름(N): game
파일 형식(T): unity (*.unity)
폴더 숨기기
저장(S)
취소

장면 저장

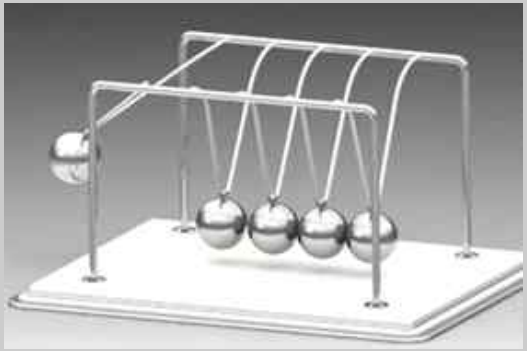

Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game

Unity Project Folder Structure

- 유니티의 폴더는 자동 관리됨
  - 임의로 변경하면 문제가 발생할 우려
- Assets
  - 대다수의 자원 파일들이 들어가는 곳
  - 절대로 외부의 파일을 임의로 넣거나 파일 탐색기로 수정해서는 안되는 위치
- Library
  - 각종 메타 데이터와 캐시 파일들이 생성되는 곳
  - 삭제해도 무방, 다시 생성 됨
- ProjectSettings
  - 각종 설정들이 저장된 장소

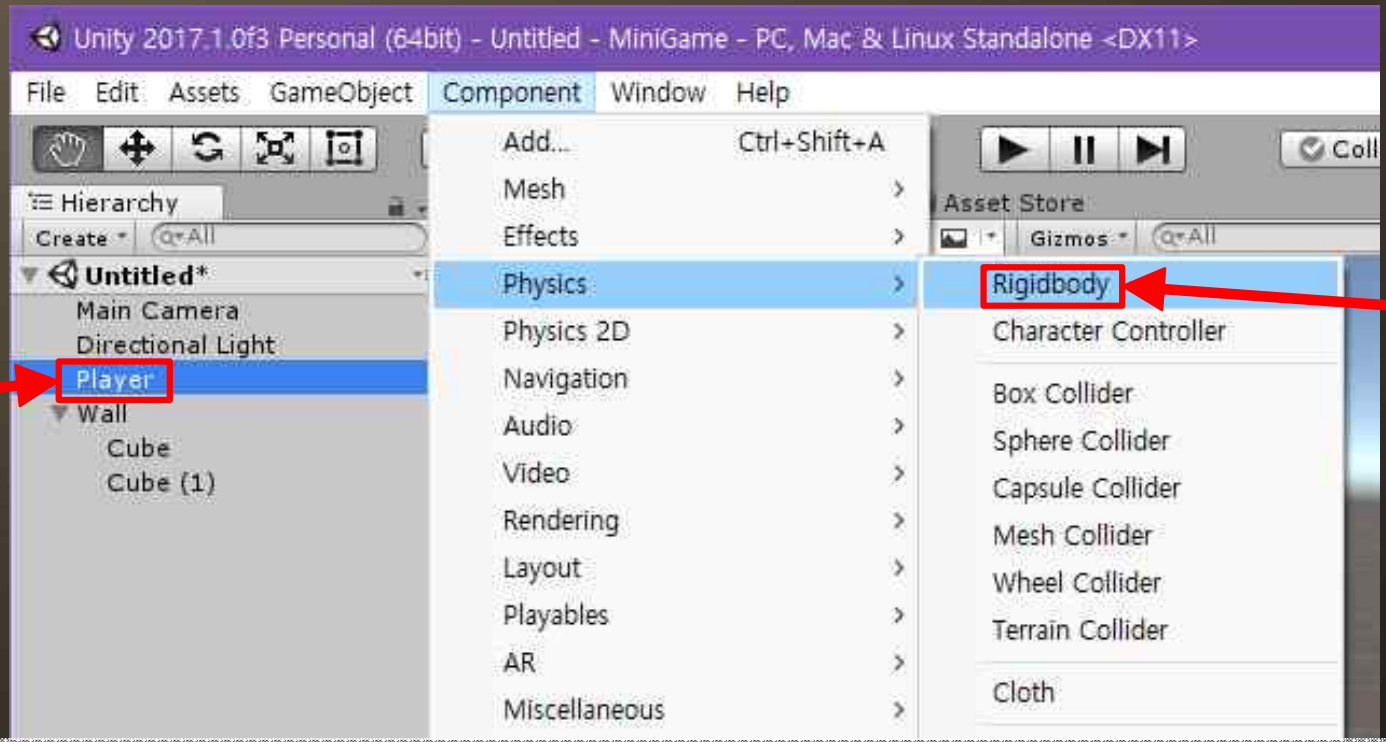


물리 현상



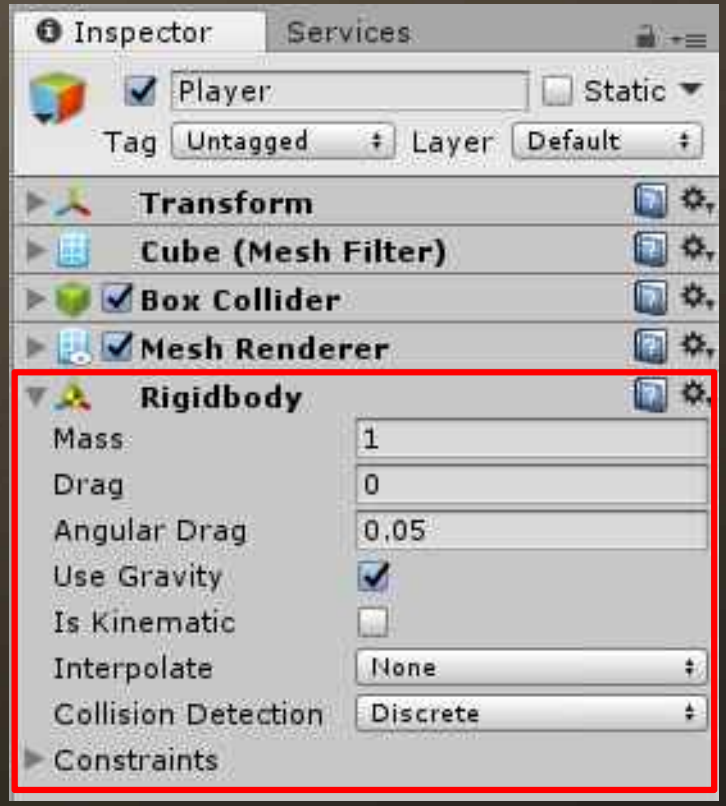
Physics

물체에 물리적 강체 요소 추가

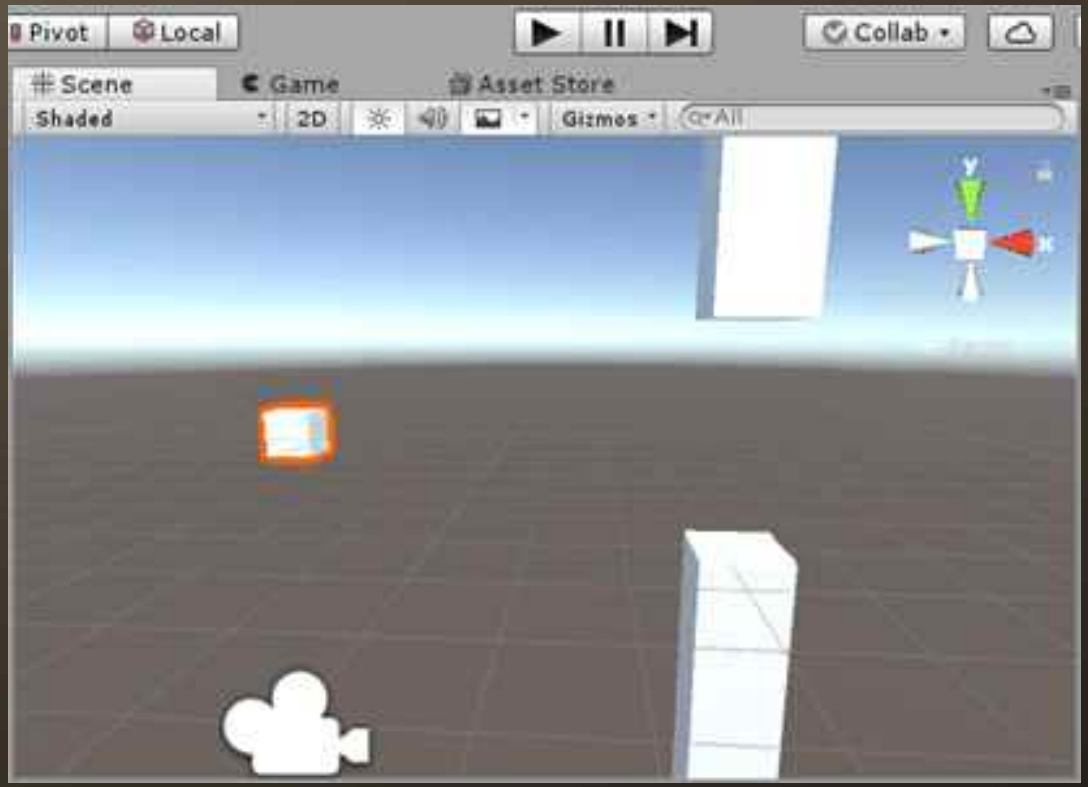


Menu: Component > Physics > Rigidbody

물체에 물리적 강체 요소 추가



플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x

물체 (Body)

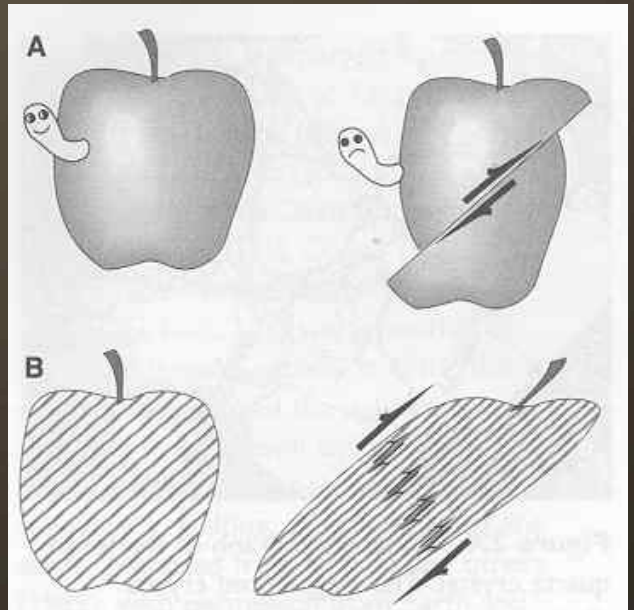
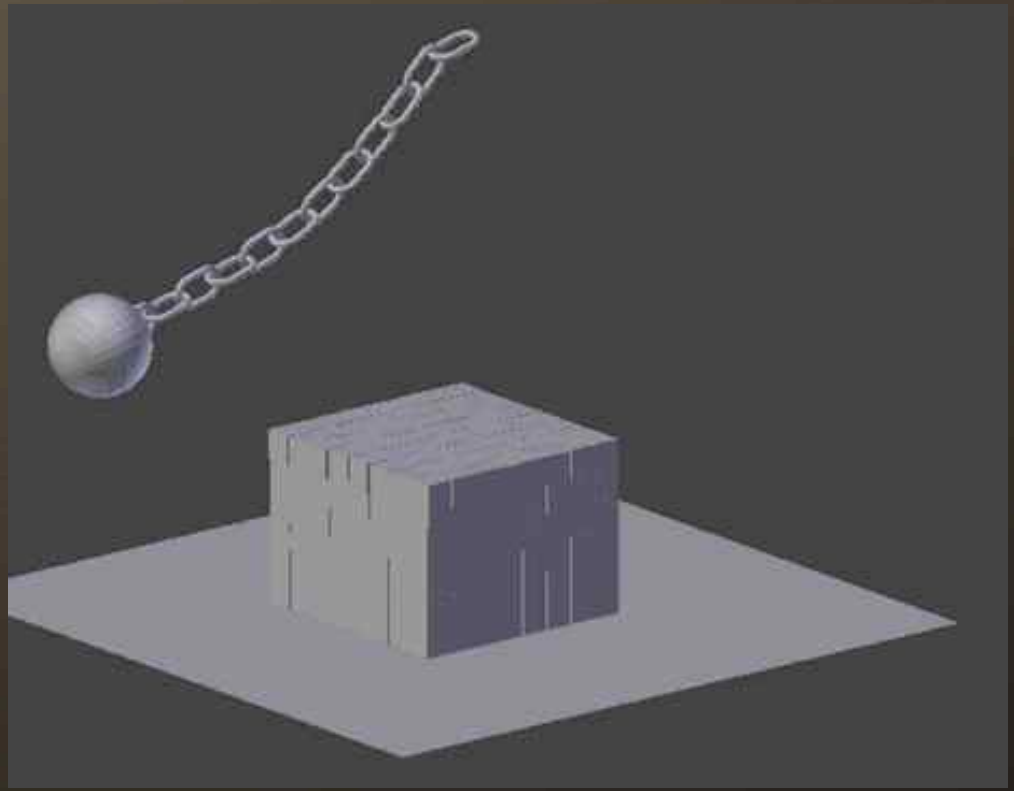
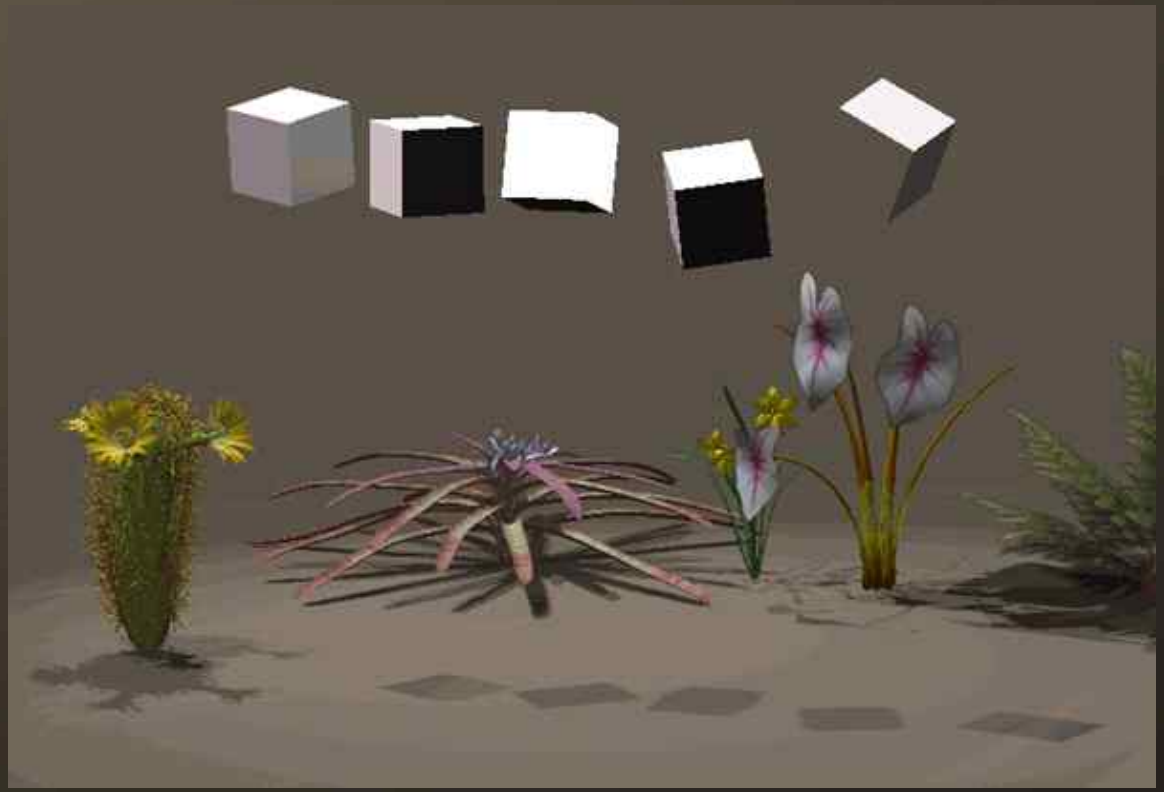


Figure 2.6 (*A*) Rigid versus (*B*) non–rigid deformation of objects by faulting is partly a matter of the closeness of spacing of the structures within the chosen field of view.

강체 (Rigidbody)



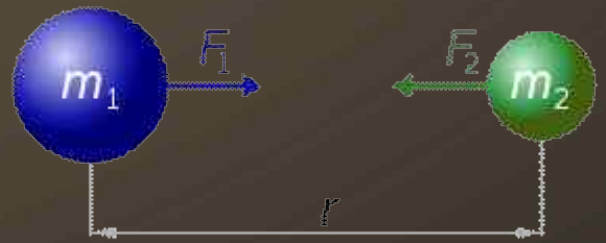
강체 (Rigidbody)



연체 (Softbody)

연체 (Softbody)

만유인력



$$F_1 = F_2 = G\frac{m_1 \times m_2}{r^2}$$

모든 점질량은 두 점을 가로지르는 선을 따라 다른 모든 점질량을 힘으로 끌어 당긴다. 이 힘은 두 상호작용하는 점질량 사이의 질량의 곱에 비례하며, 두 점질량 사이의 거리에는 제곱에 반비례한다.

중력가속도

$$mg = G\frac{Mm}{R^2}$$

물리학에서 중력에 의해 운동하는 물체가 지니는 가속도이다.
지오이드를 기준으로 한 지구의 표준 중력가속도 값은 9.80665 m/s^2이다.

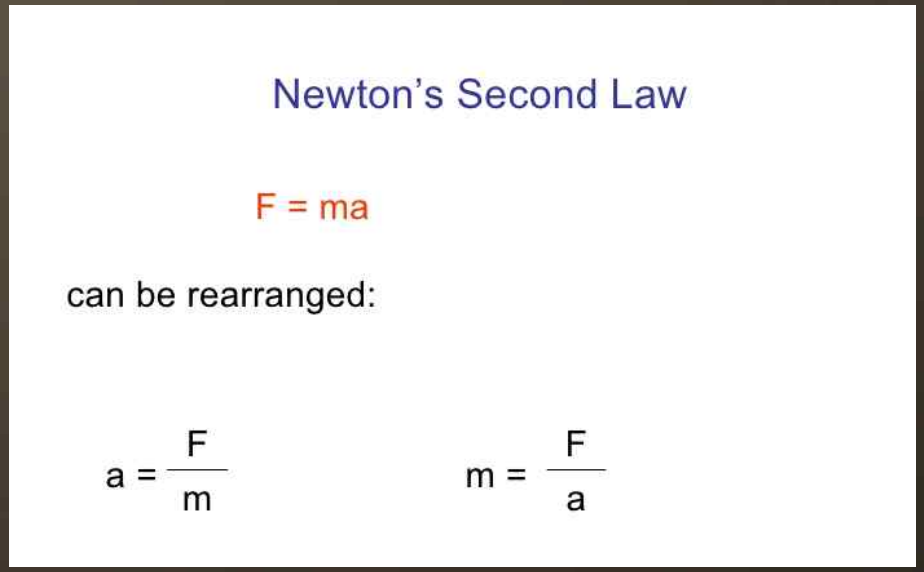
무게

$F_g = mg = W$

Weight is another word for the force of gravity

무게는 물체의 질량에 중력가속도를 곱한 값이다.
중력이 작용하는 모든 강체에 적용되는 힘이다.

힘과 가속도



가속도를 구하려면 힘을 질량으로 나누면 된다.
게임에서는 0으로 나누는 것을 예방하기 위해 일반적으로 1/m(inverseMass)로 질량을 표현하며 이 값을 0으로 만들어서 움직이지 않는 물체를 구현할 수 있

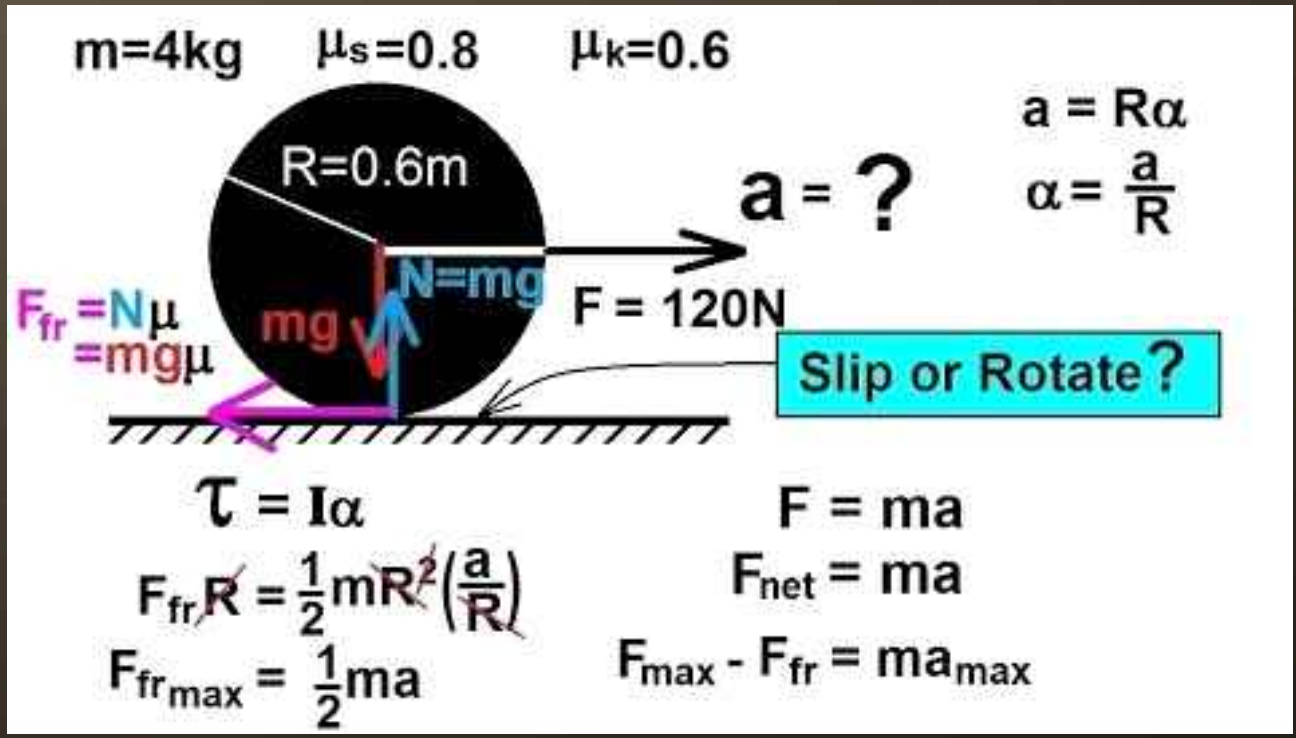
강체의 위치 변화

$$x = x_o + v_o t + \frac{1}{2} a t^2$$

강체의 속도 변화

$$v = v_o + at$$

회전운동


m=4kg
μ_s=0.8
μ_k=0.6
R=0.6m
a = ?
a = Rα
α = a/R
N=mg
mg
F_{fr} =Nμ
=mgμ
F = 120N
Slip or Rotate?
τ = Iα
$F_{fr}R = \frac{1}{2}mR^2\left(\frac{a}{R}\right)$
$F_{fr\,max} = \frac{1}{2}ma$
F = ma
F_{net} = ma
$F_{max} - F_{fr} = ma_{max}$

물리

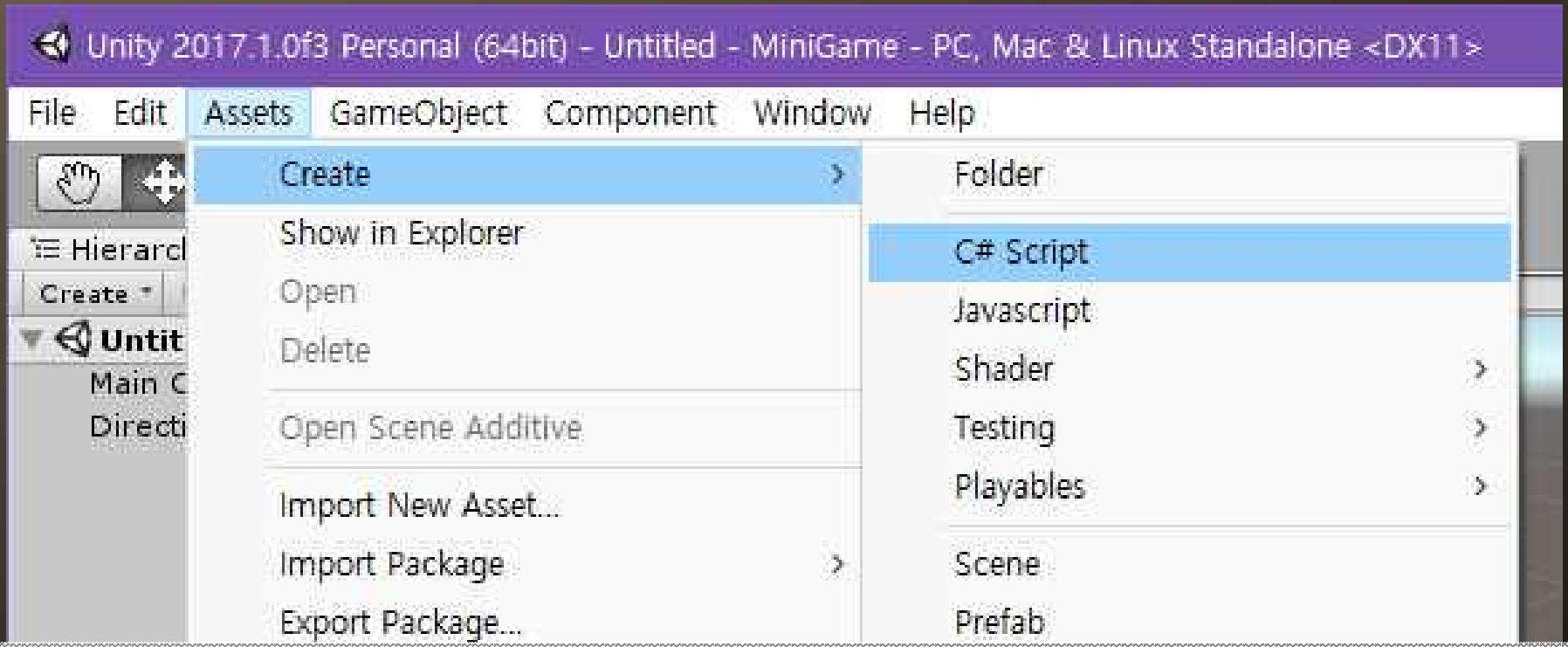
물리 엔진

. 걱정 마세요. 물리 엔진에게 맡기세요.

스크립트

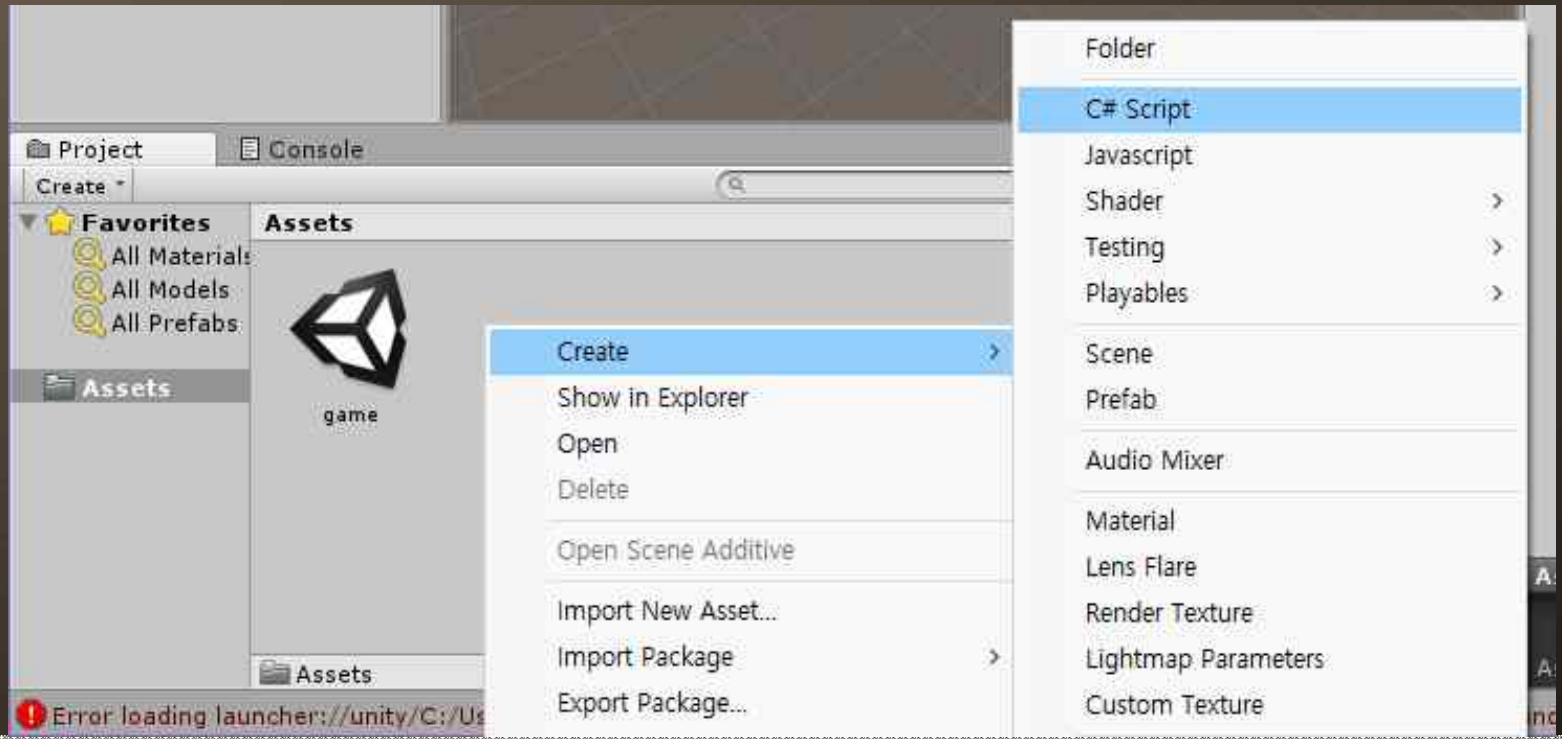
Scripts

스크립트 추가

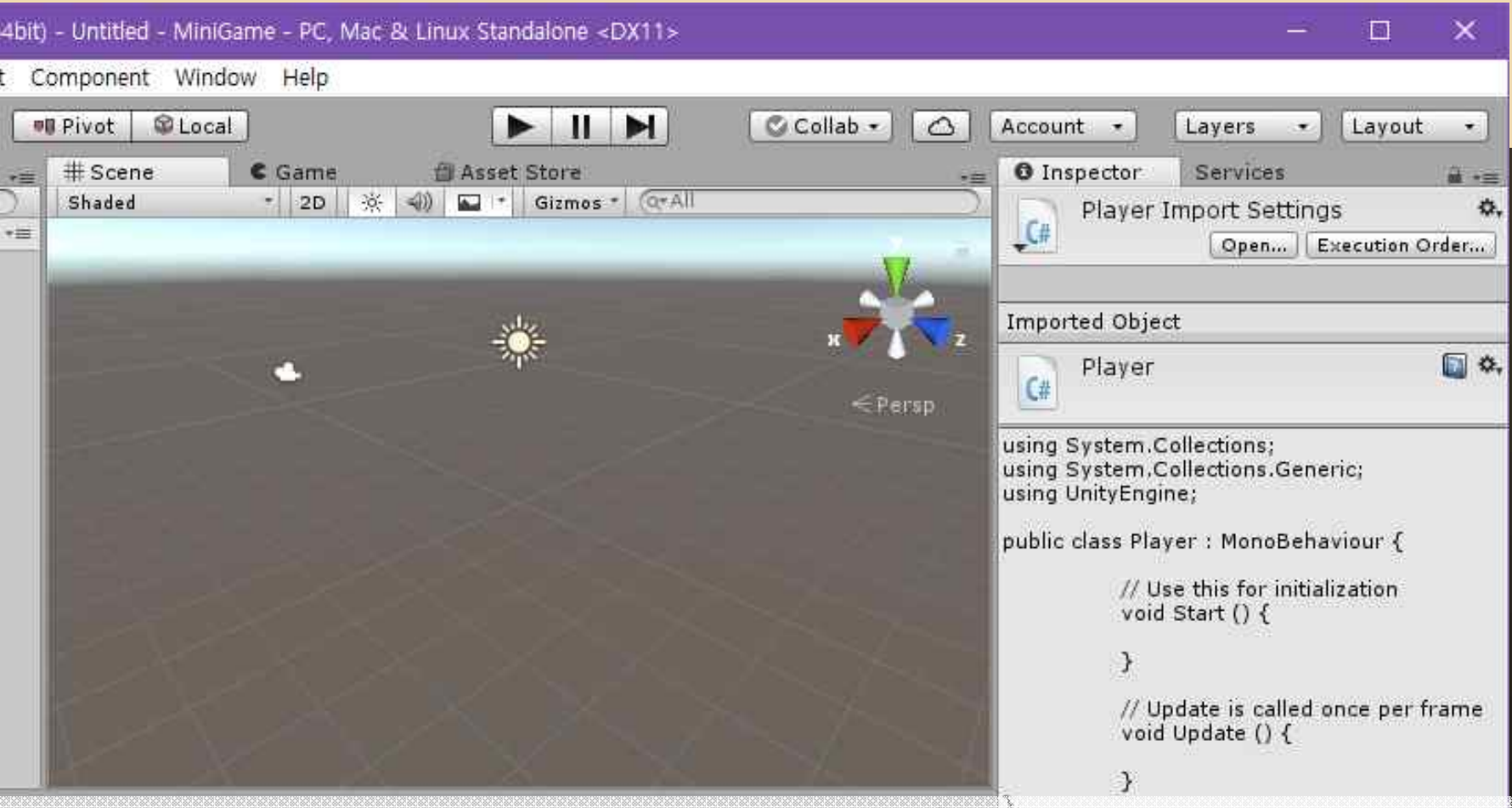


Menu: Assets > Create > C# Script

스크립트 추가

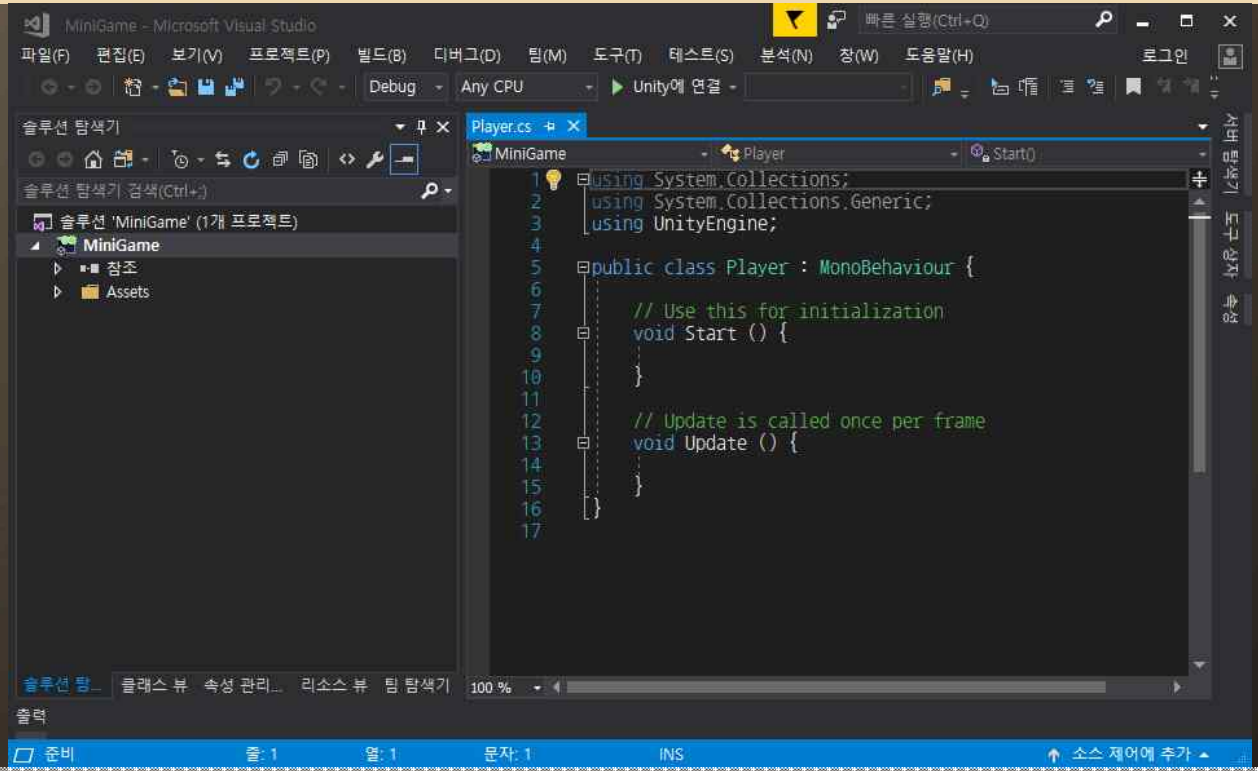


Menu: Assets > Create > C# Script



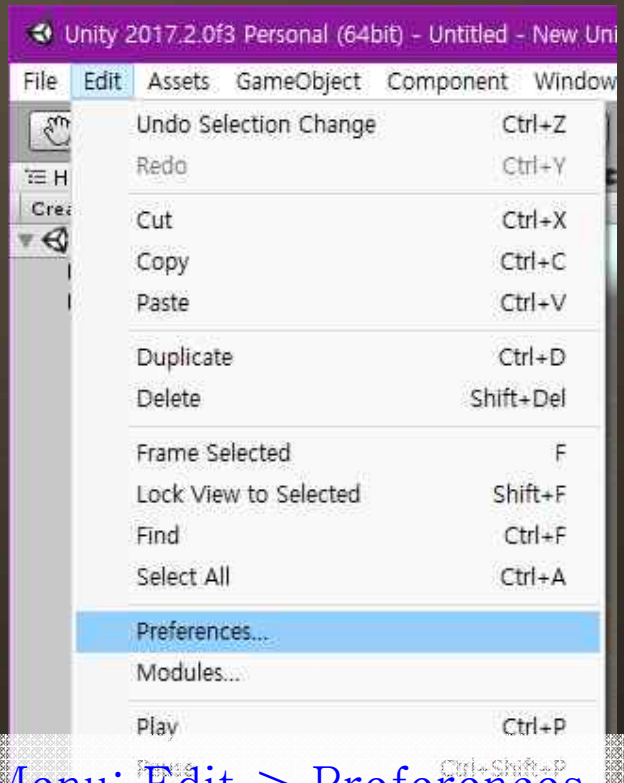
Class name: “Player”

스크립트 편집



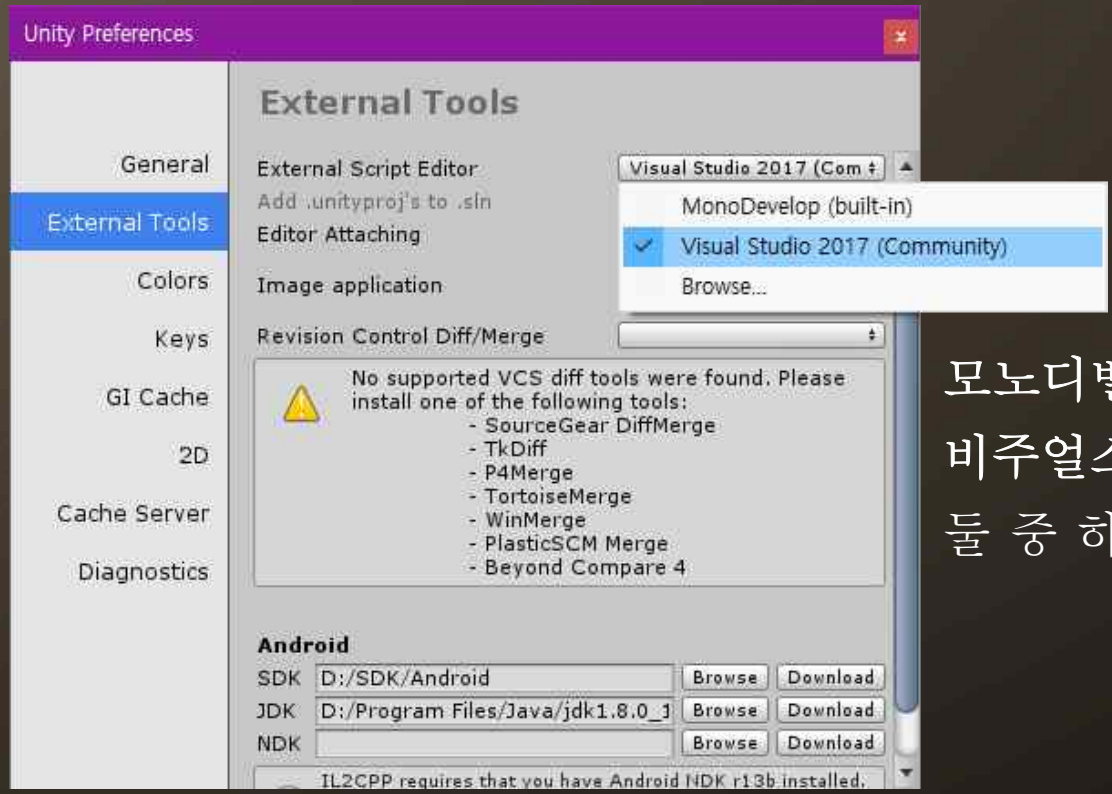
스크립트를 더블 클릭하면 비주얼 스튜디오 실행

스크립트 편집기 선택



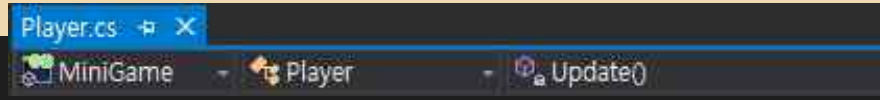
Menu: Edit > Preferences...

스크립트 편집기 선택



모노디벨롭과
비주얼스튜디오
둘 중 하나로 선택 가능

점프 기능 추가

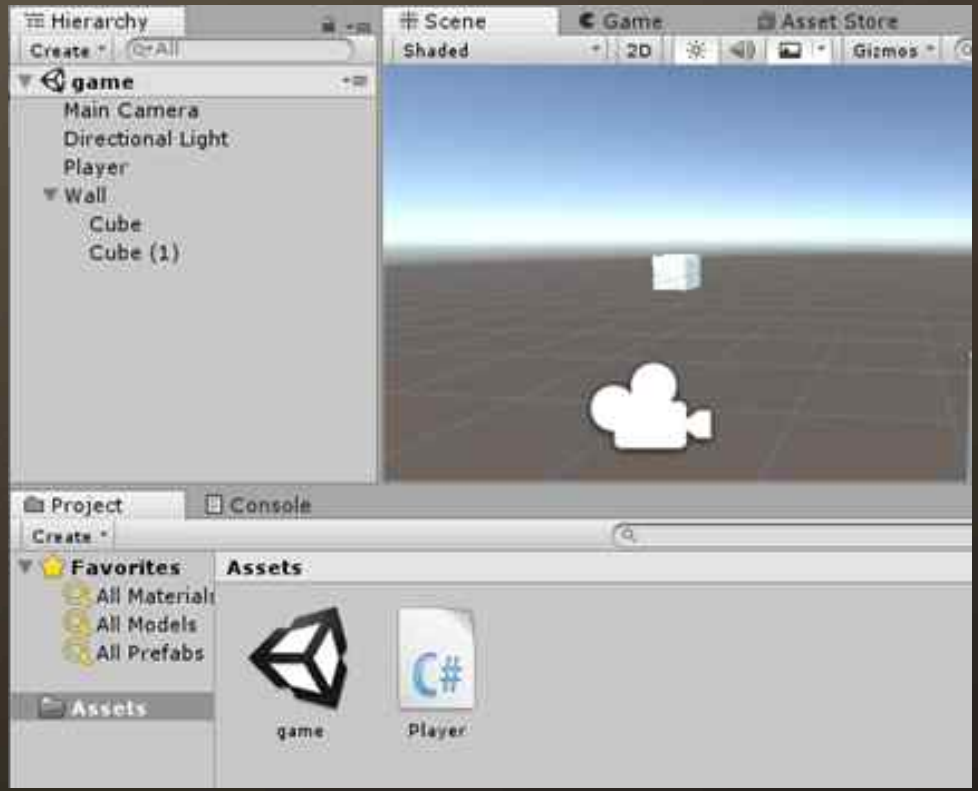



```
public class Player : MonoBehaviour {

    public float jumpPower = 5;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        if (Input.GetButtonDown("Jump"))
```

스크립트 저장

Menu: File > Save (Ctrl+S)

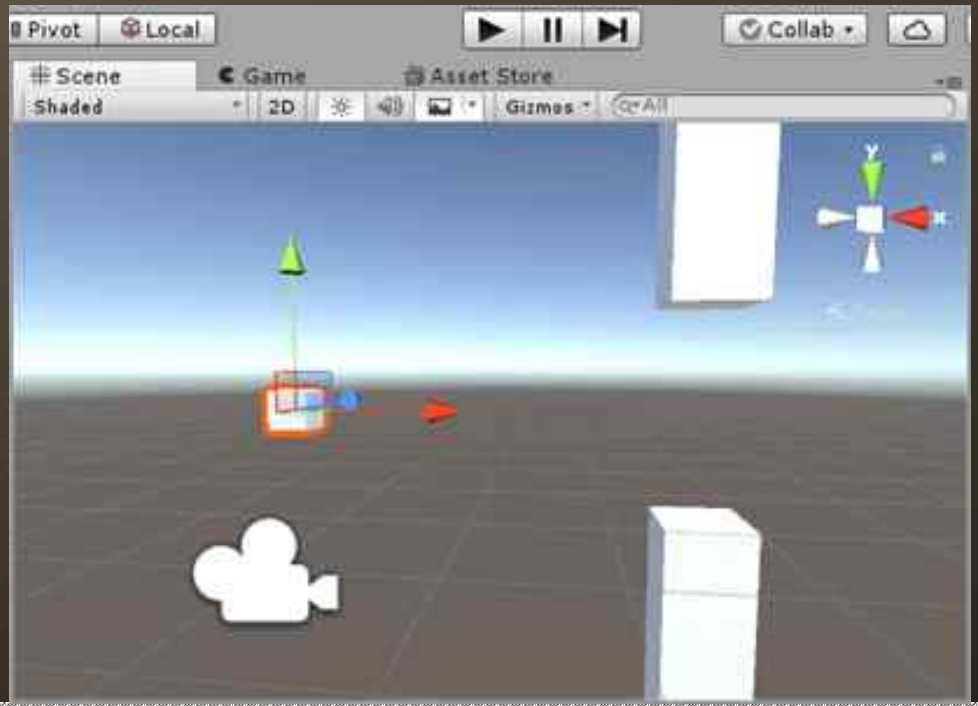
스크립트 연결


Hierarchy
Create
All
game
Main Camera
Directional Light
Player
Wall
Cube
Cube (1)
Scene
Game
Asset Store
Shaded
2D
Gizmos
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player

스크립트 연결

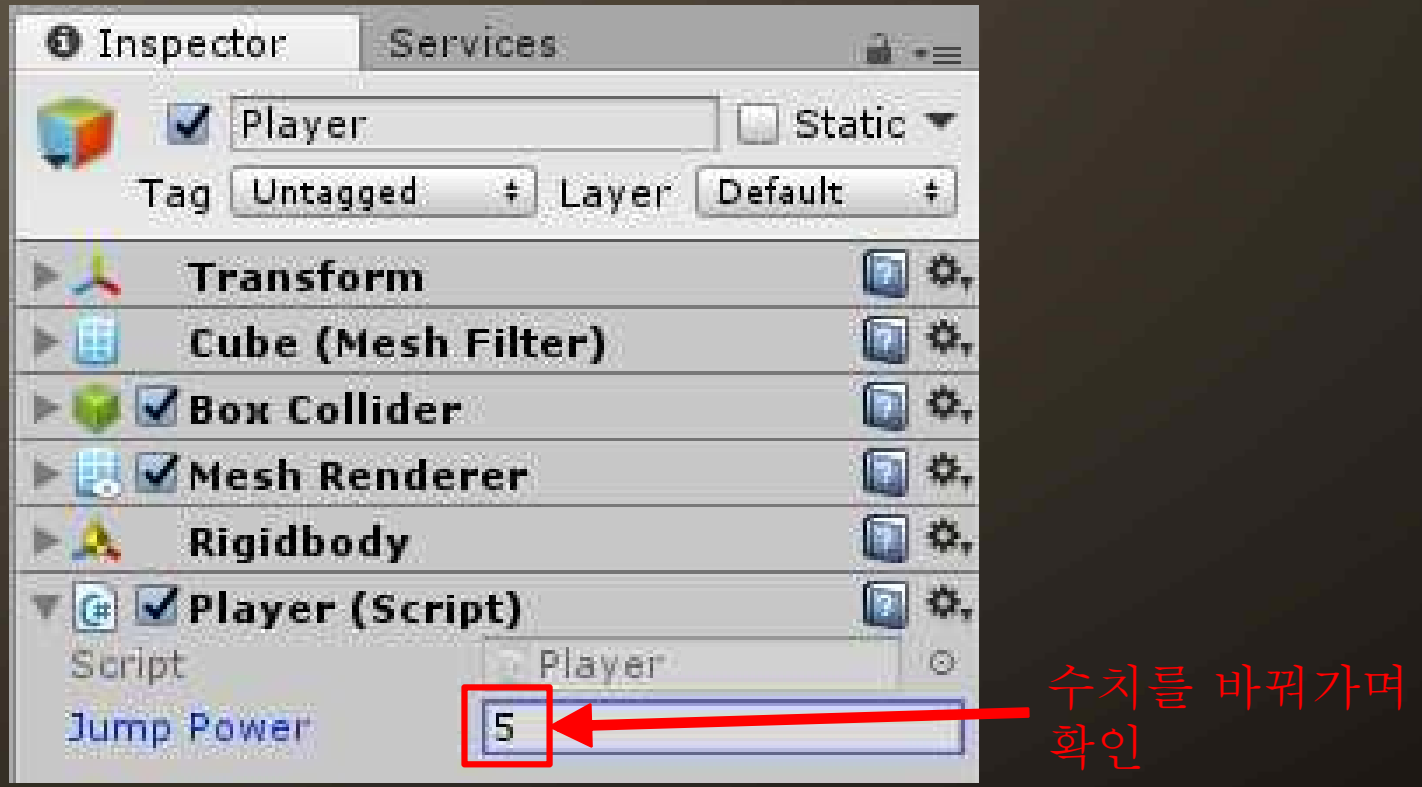

Inspector
Services
Player
Static
Tag
Untagged
Layer
Default
Transform
Cube (Mesh Filter)
Box Collider
Mesh Renderer
Rigidbody
Player (Script)
Script
Player
Jump Power
5

플레이

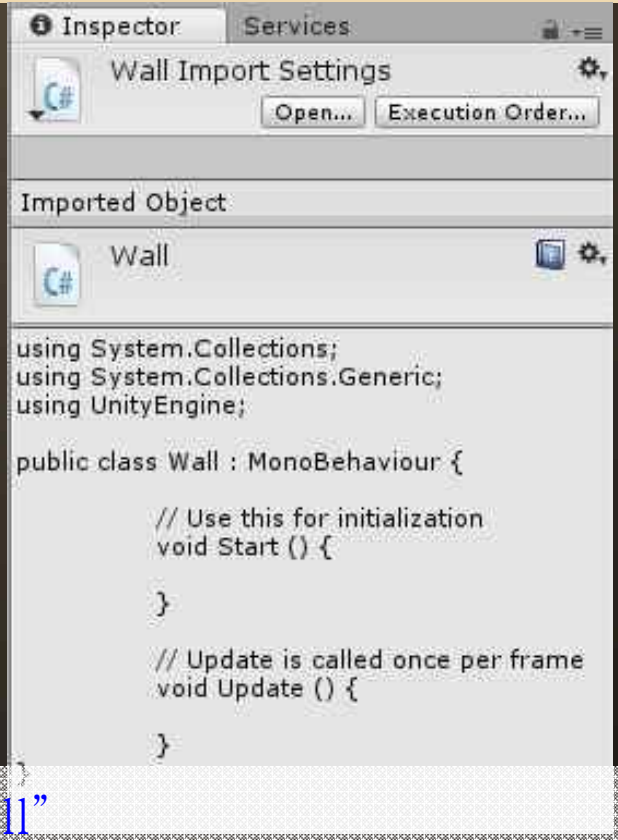


Space 키로 점프 확인

스크립트 연결


Inspector
Services
Player
Static
Tag
Untagged
Layer
Default
Transform
Cube (Mesh Filter)
Box Collider
Mesh Renderer
Rigidbody
Player (Script)
Script
Player
Jump Power
5
수치를 바꿔가며
확인

스크립트 추가



Class name: "Wall"

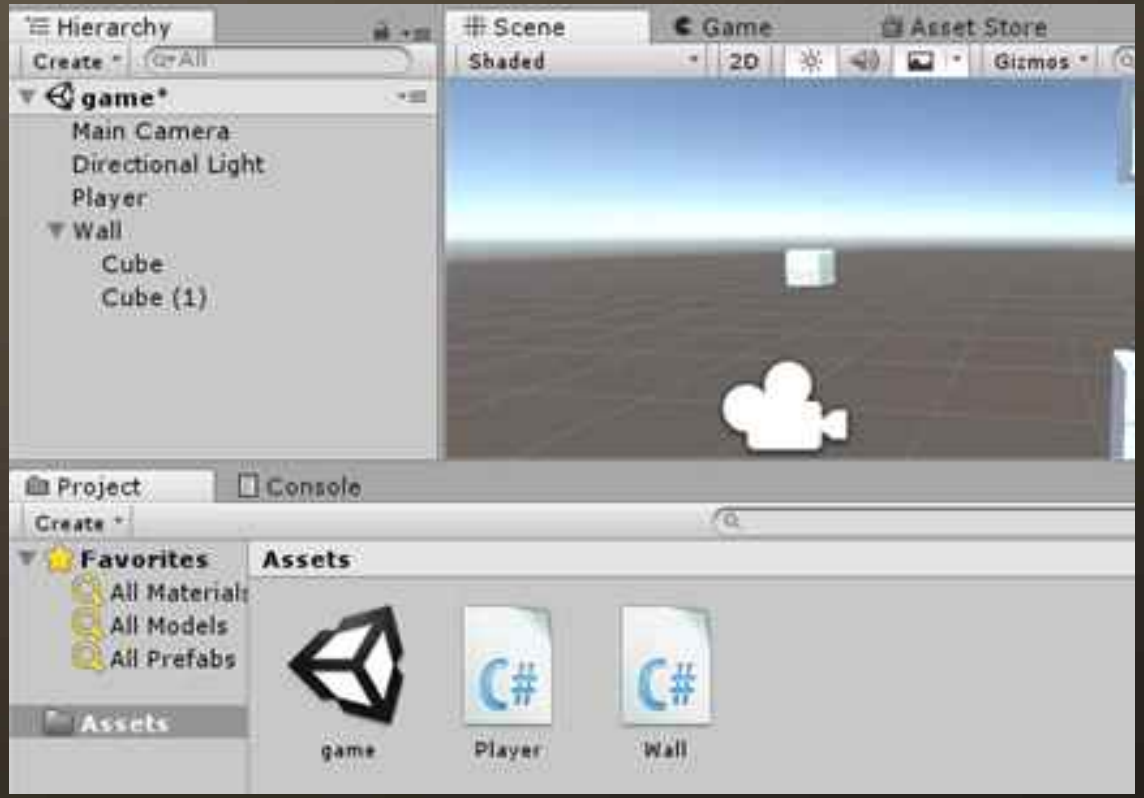
벽 이동 추가

Wall.cs

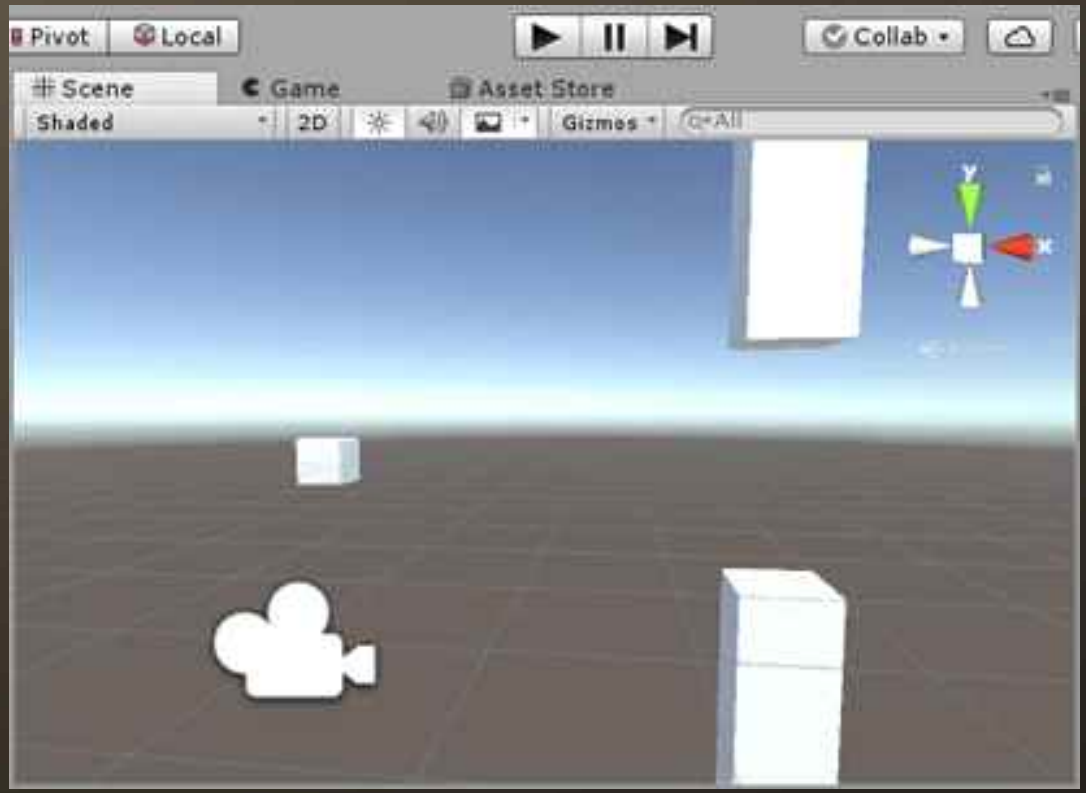
MiniGame | Wall | Update()

```
public class Wall : MonoBehaviour {

        public float speed = -5;

        // Use this for initialization
        void Start () {


        }

        // Update is called once per frame
        void Update () {
                transform.Translate(speed * Time.deltaTime, 0, 0);
        }
```

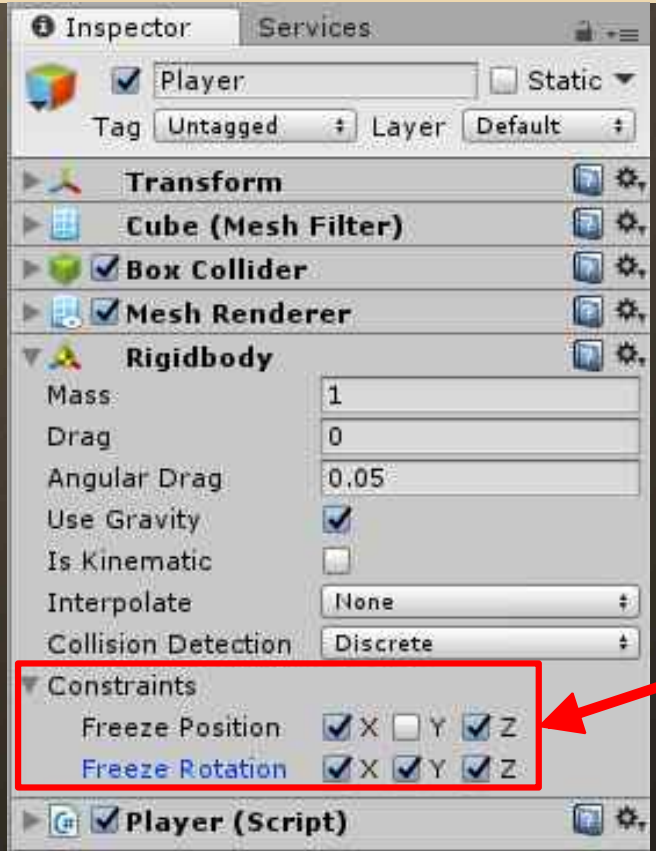
스크립트 연결


Hierarchy
Create
game*
Main Camera
Directional Light
Player
Wall
Cube
Cube (1)
Scene
Game
Asset Store
Shaded
2D
Gizmos
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Wall

플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
Y

플레이어 큐브 고정시키기

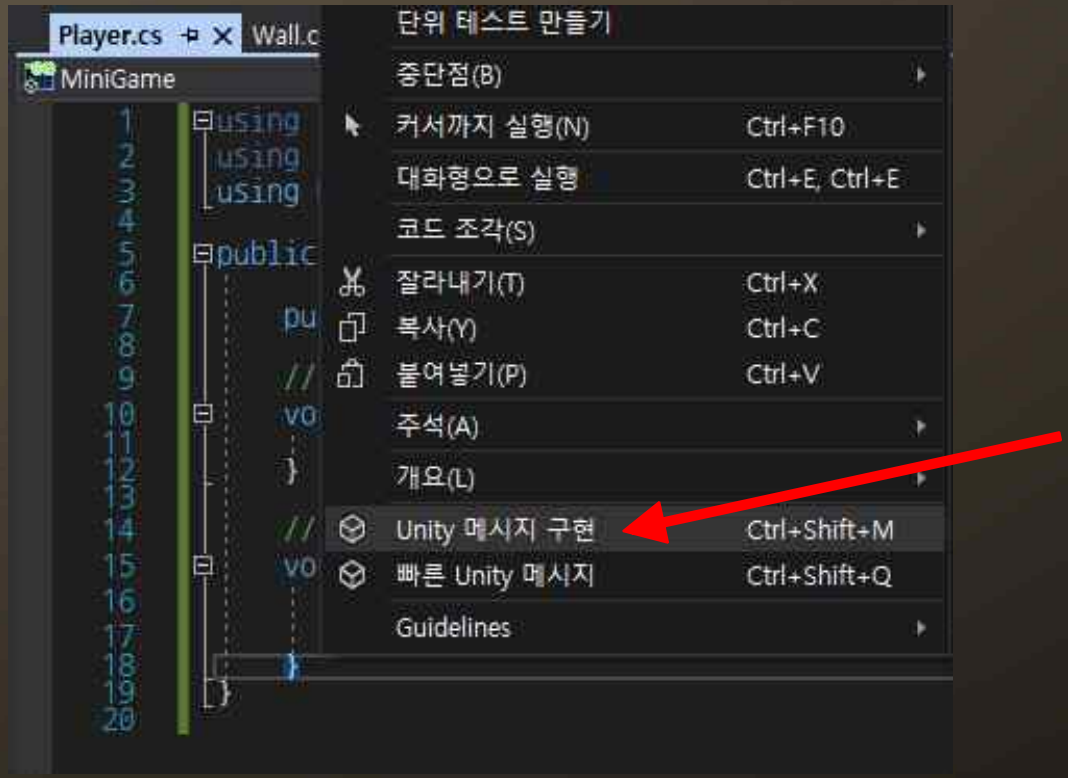


Y축 위치 이동을 제외하고
전부 변화가 없도록 고정

충돌 처리 추가

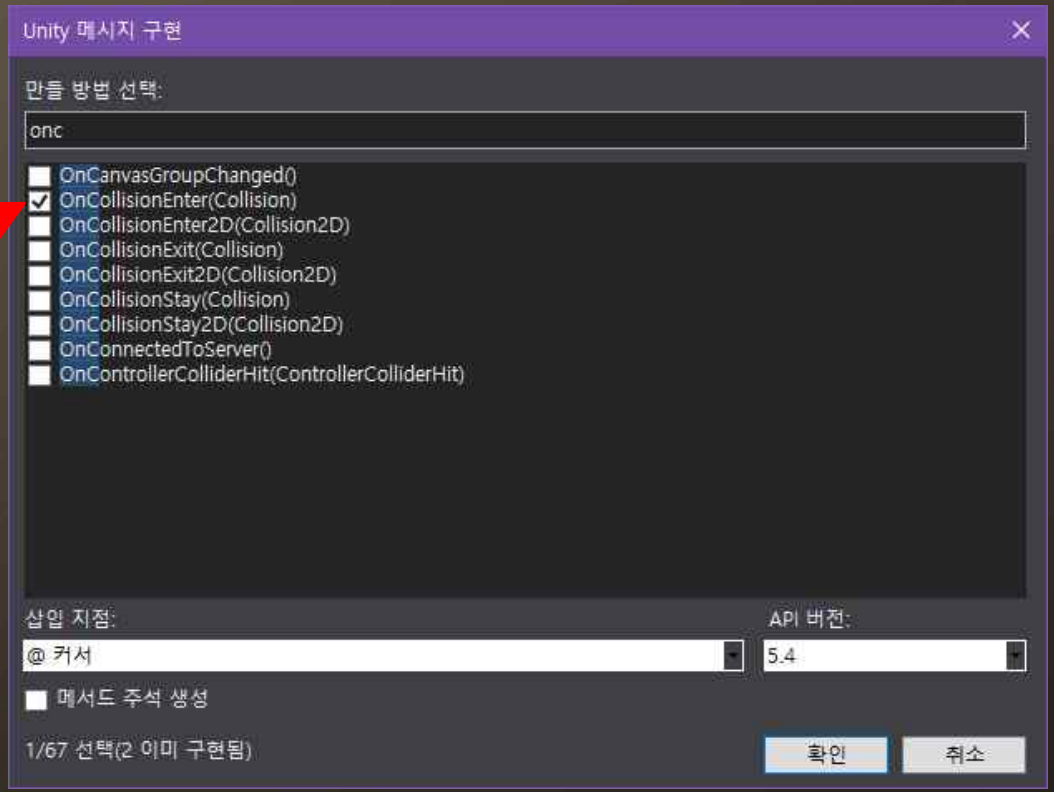
- Player.cs 스크립트 수정
  - 플레이어가 벽에 충돌하면 처음부터 다시 시작하기
  - Visual Studio 편집 화면에서 오른쪽 클릭
  - Unity 메세지 구현 메뉴를 실행

충돌 처리 추가

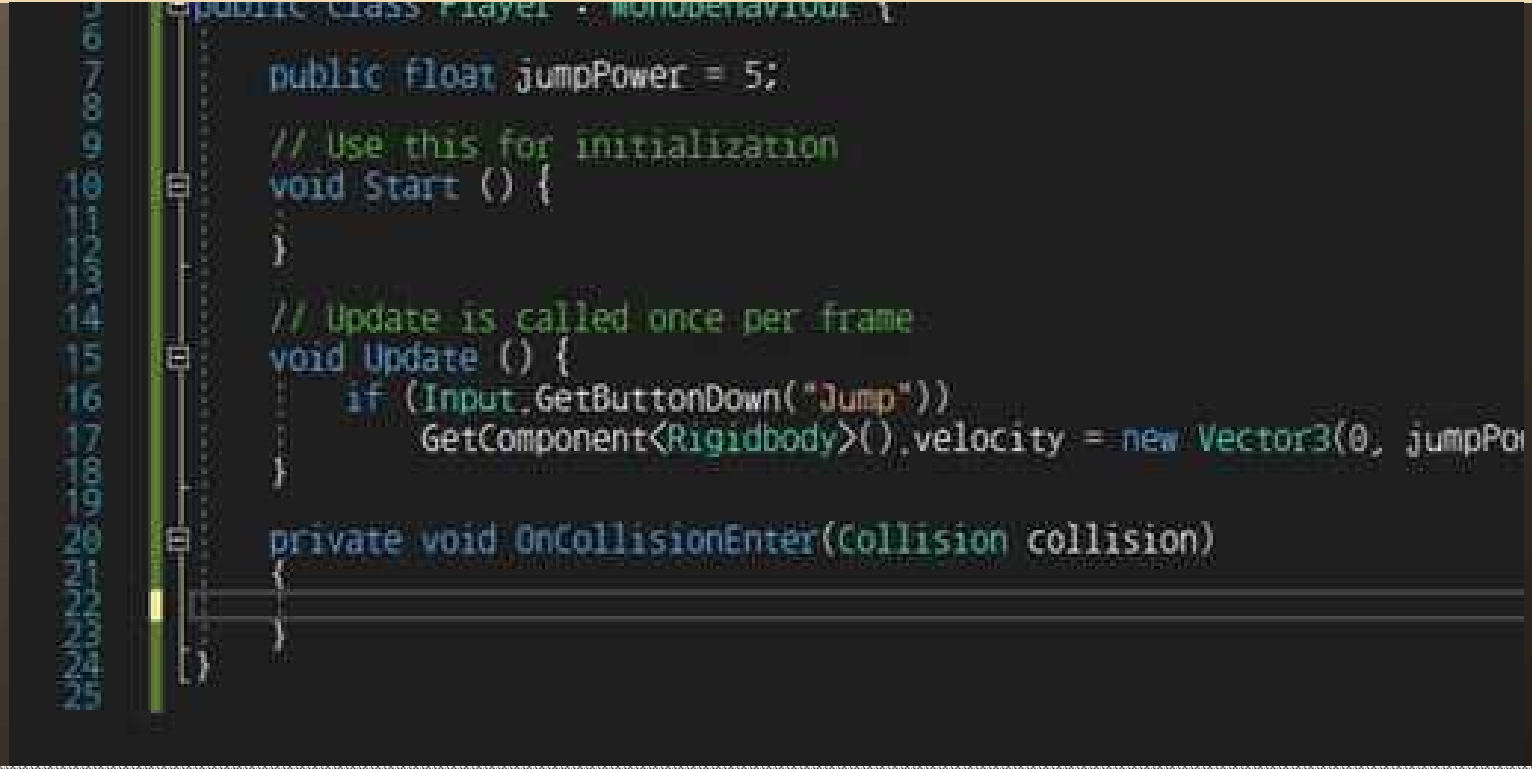

Player.cs
Wall.c
MiniGame
using
using
using
public
pu
//
vo
}
//
vo
}
}
단위 테스트 만들기
중단점(B)
커서까지 실행(N) Ctrl+F10
대화형으로 실행 Ctrl+E, Ctrl+E
코드 조각(S)
잘라내기(T) Ctrl+X
복사(Y) Ctrl+C
붙여넣기(P) Ctrl+V
주석(A)
개요(L)
Unity 메시지 구현 Ctrl+Shift+M
빠른 Unity 메시지 Ctrl+Shift+Q
Guidelines

충돌 처리 추가

OnCollisionEnter
메세지 체크

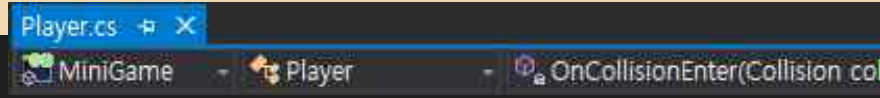


스크립트 편집



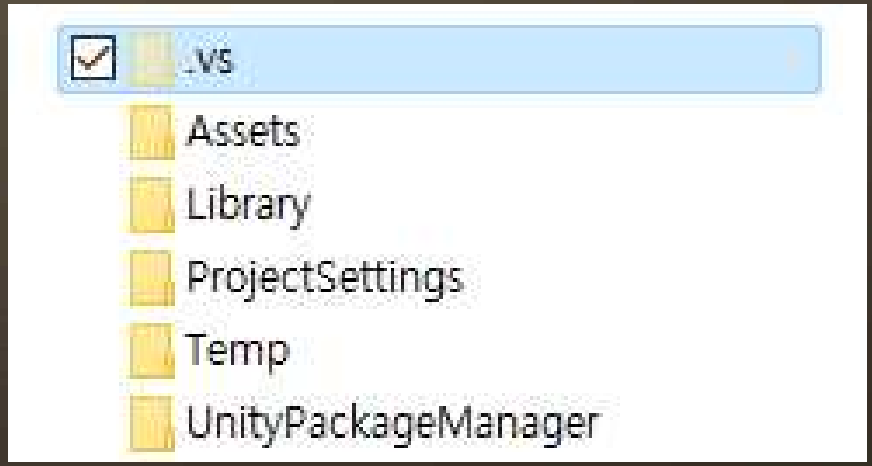
전구 아이콘으로 using을 자동으로 추가하고, Tab 키를 활용하여 코드 자동 완성

새로 로딩 추가




```
private void OnCollisionEnter(Collision collision)
{
        SceneManager.LoadScene(SceneManager.GetActiveScene().n
ame);
}
```

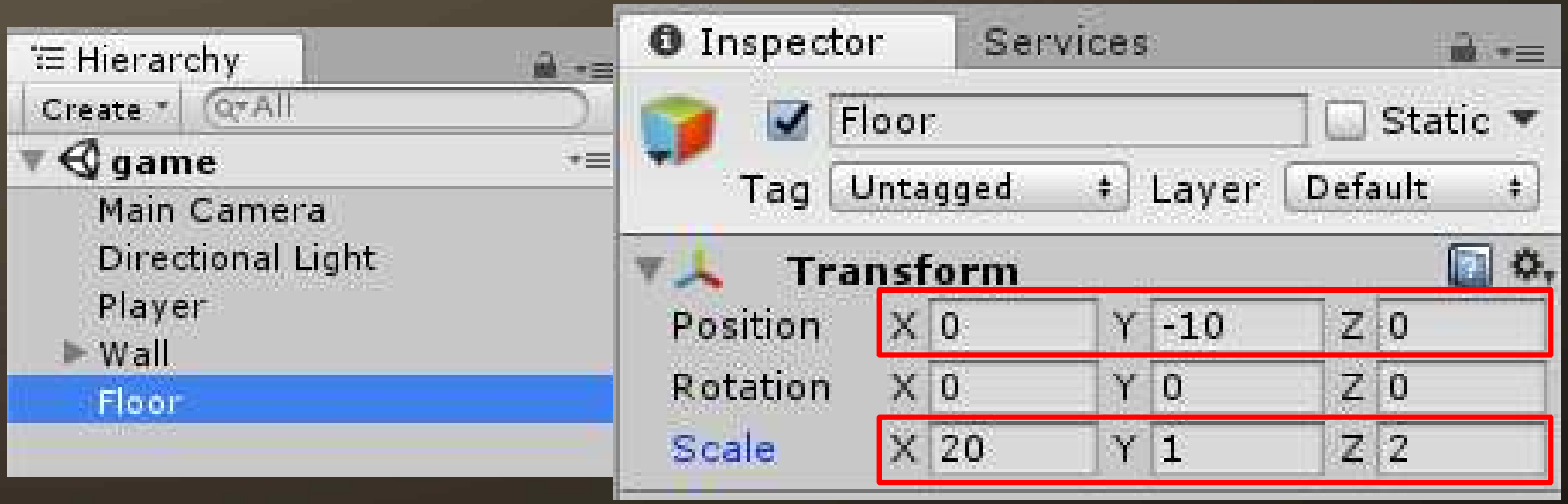
VS의 Intellisense 오류 발생시



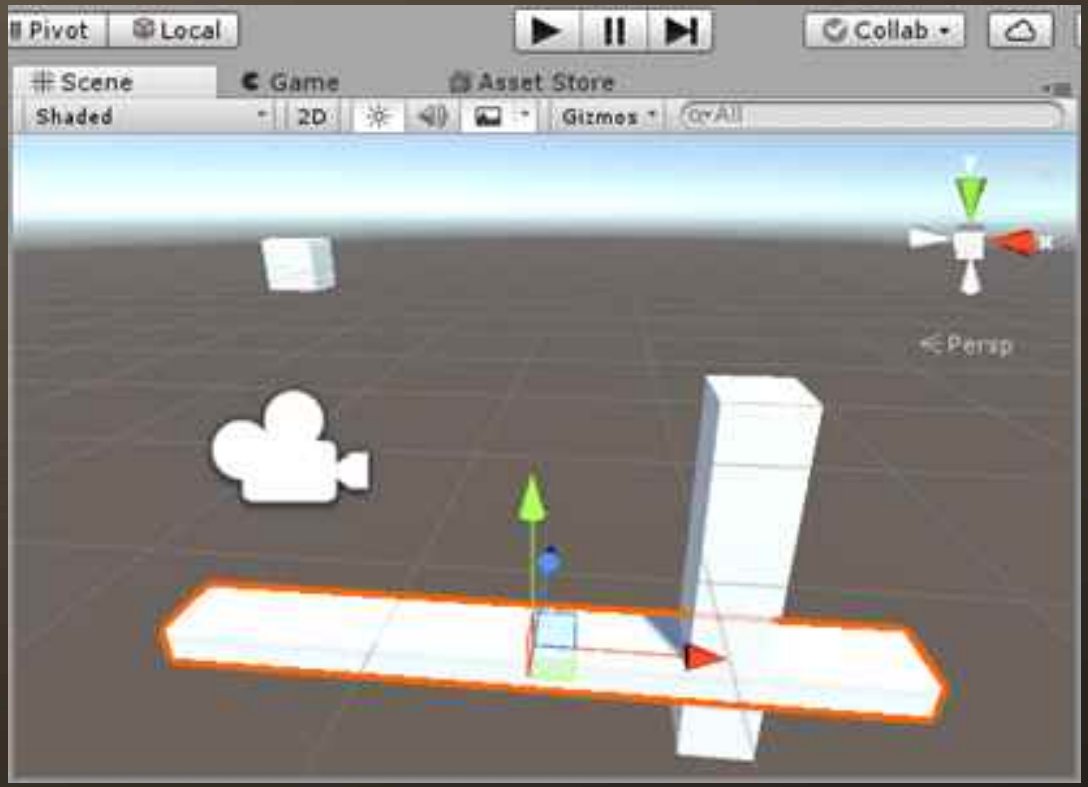
VS의 캐시를 삭제하고 다시 시작

바닥 추가

6. GameObject > 3D Object > Cube



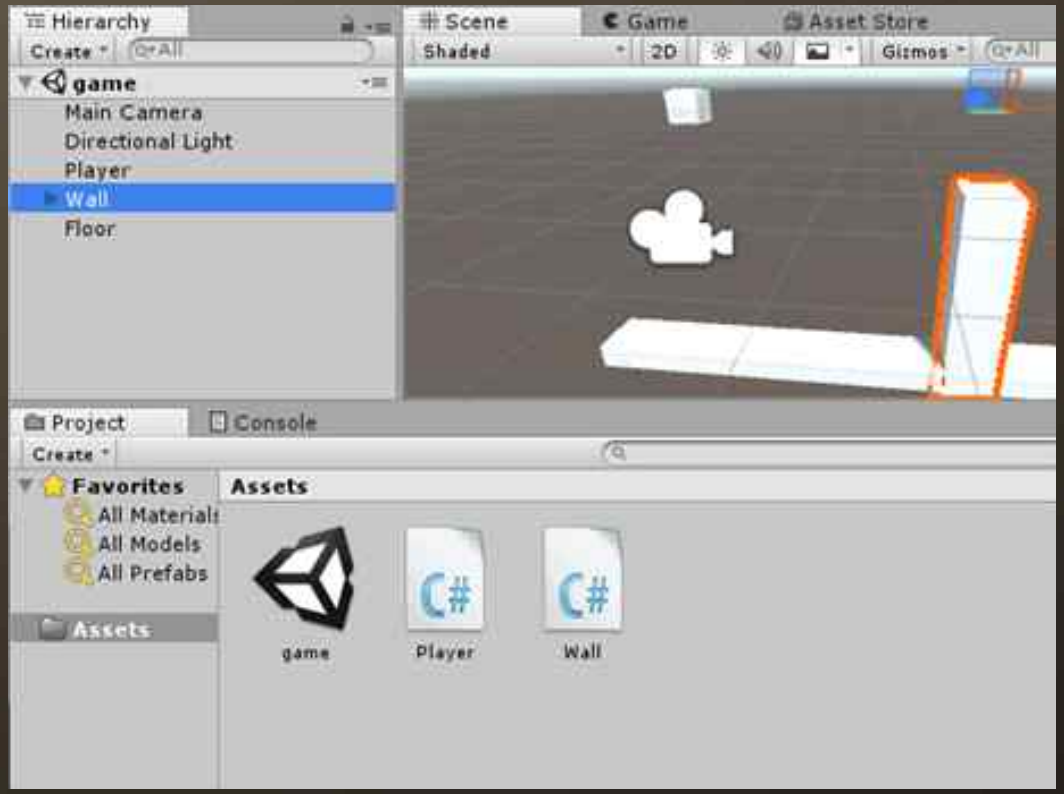
플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp

프리팹

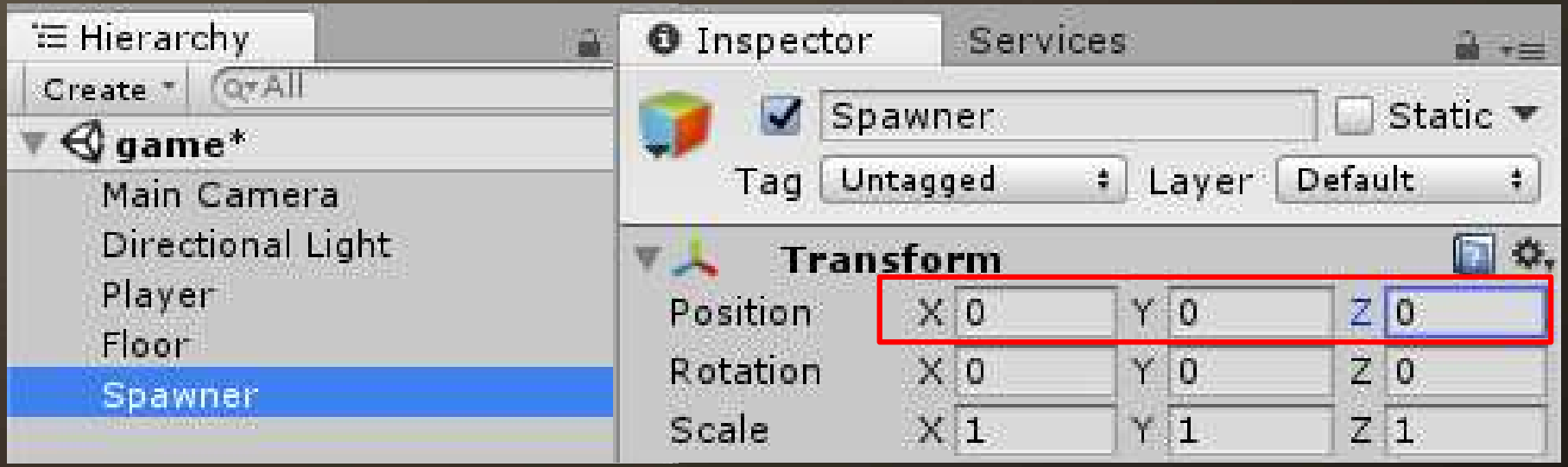
Prefabs

벽 프리팹 만들기

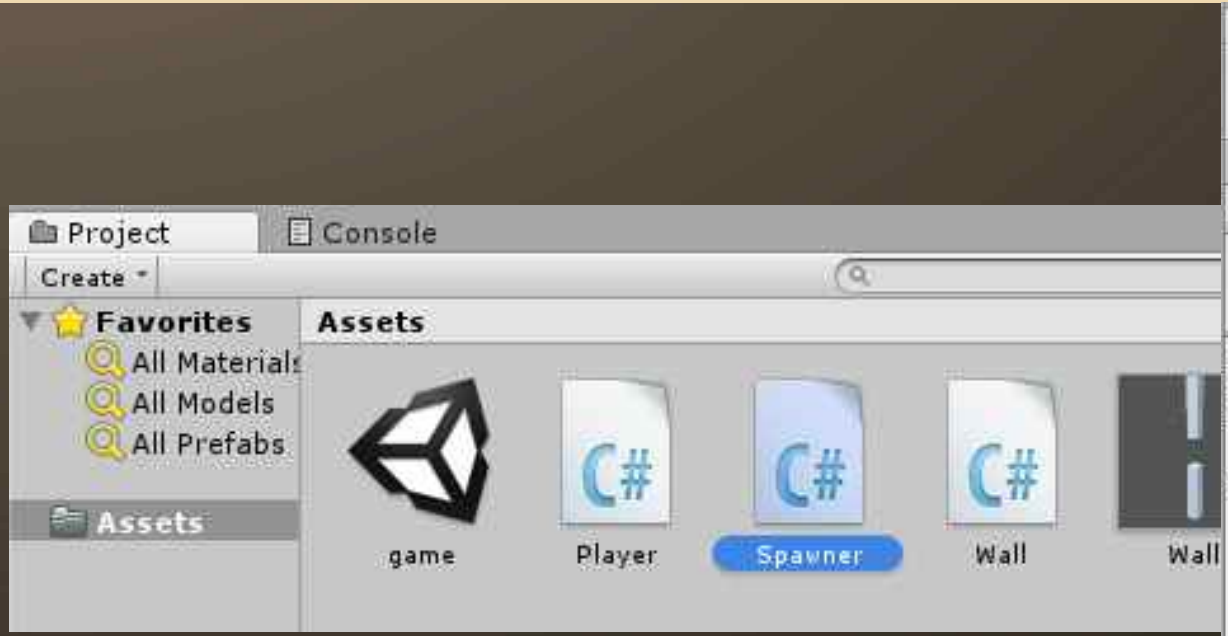

Hierarchy
Create
game
Main Camera
Directional Light
Player
Wall
Floor
Scene
Game
Asset Store
Shaded
2D
Gizmos
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Wall

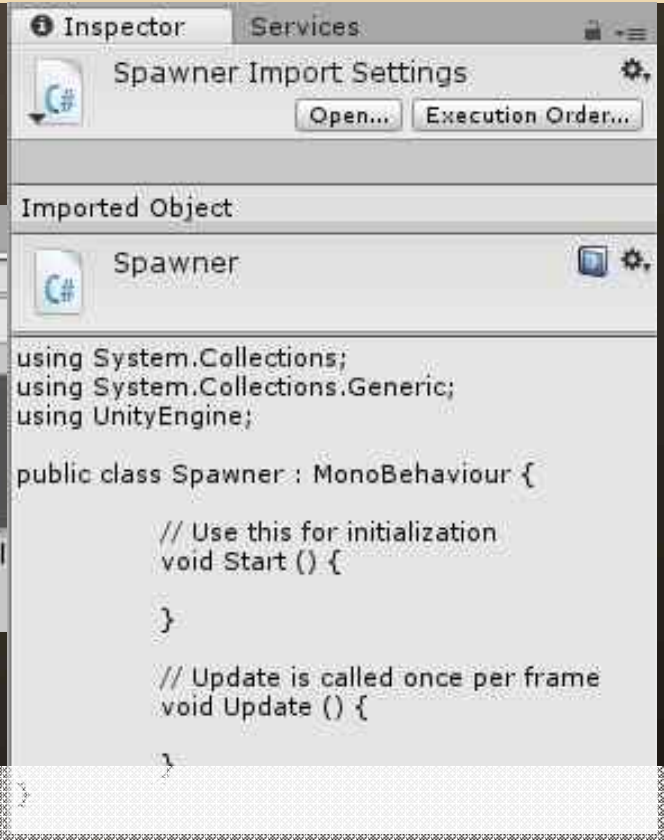
스크립트 실행용 빈 오브젝트 생성

7. GameObject > Create Empty



스크립트 추가





Class name: "Spawner"

일정 시간마다 프리팹 생성

Spawner.cs | MiniGame | Spawner | Start()

```
public class Spawner : MonoBehaviour {

    public GameObject wallPrefab;
    public float interval = 1.5f; // 일정 시간마다
    float term;

    // Use this for initialization
    void Start () {
        term = interval;  // 시작부터 벽이 하나 나오기 위해
    }
```

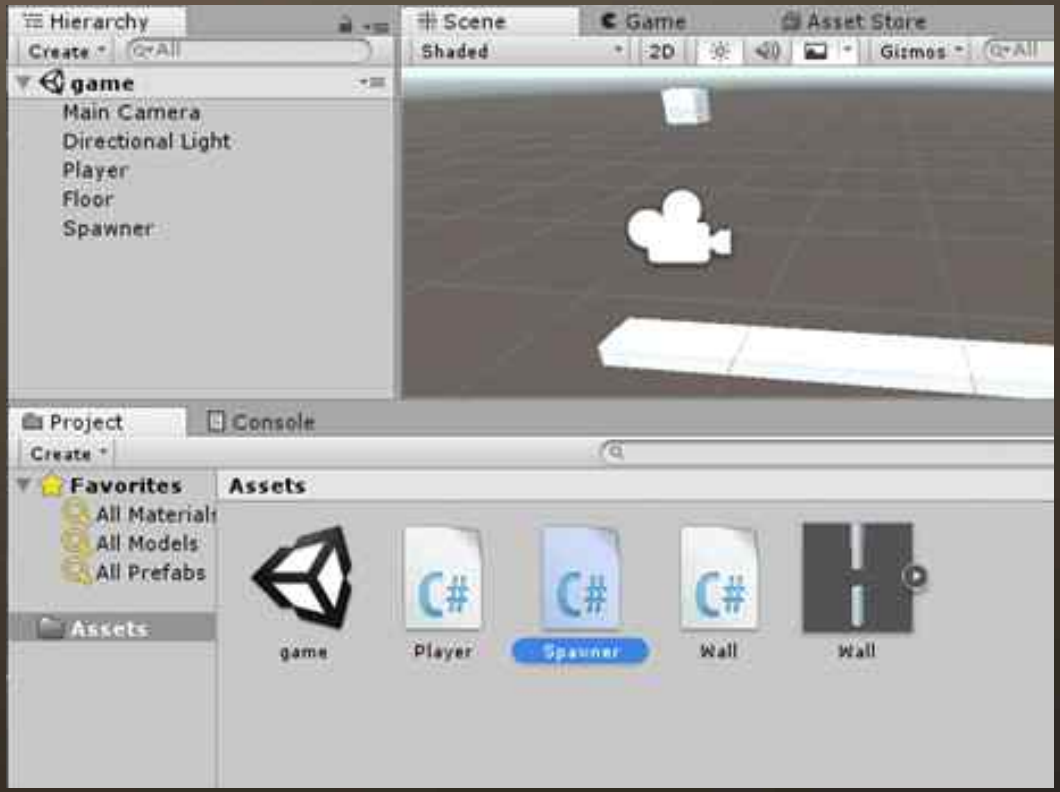
일정 시간마다 프리팹 생성

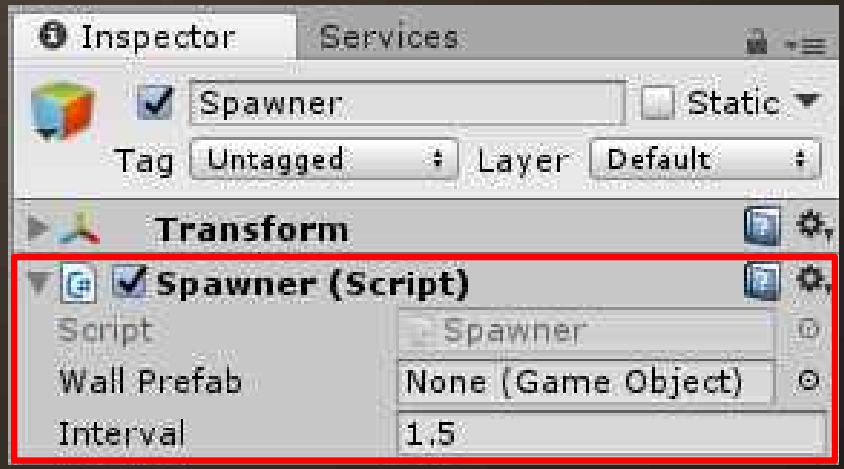
Spawner.cs

MiniGame | Spawner | Update()

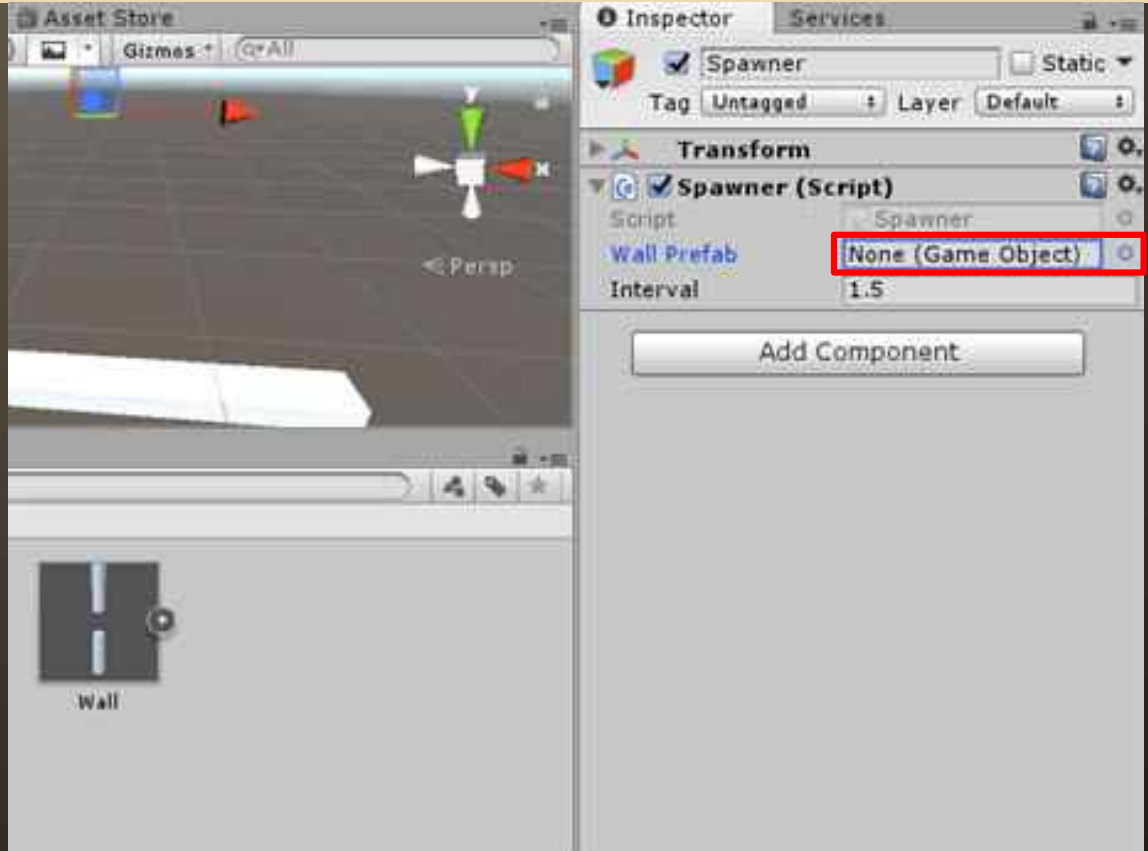
```
    // Update is called once per frame
    void Update () {
        term += Time.deltaTime;
        if (term >= interval)      // 일정 시간이 지나면
        {
            Instantiate(wallPrefab, transform.position
                , transform.rotation);
            term -= interval;
        }
    }
}
```

스크립트 연결

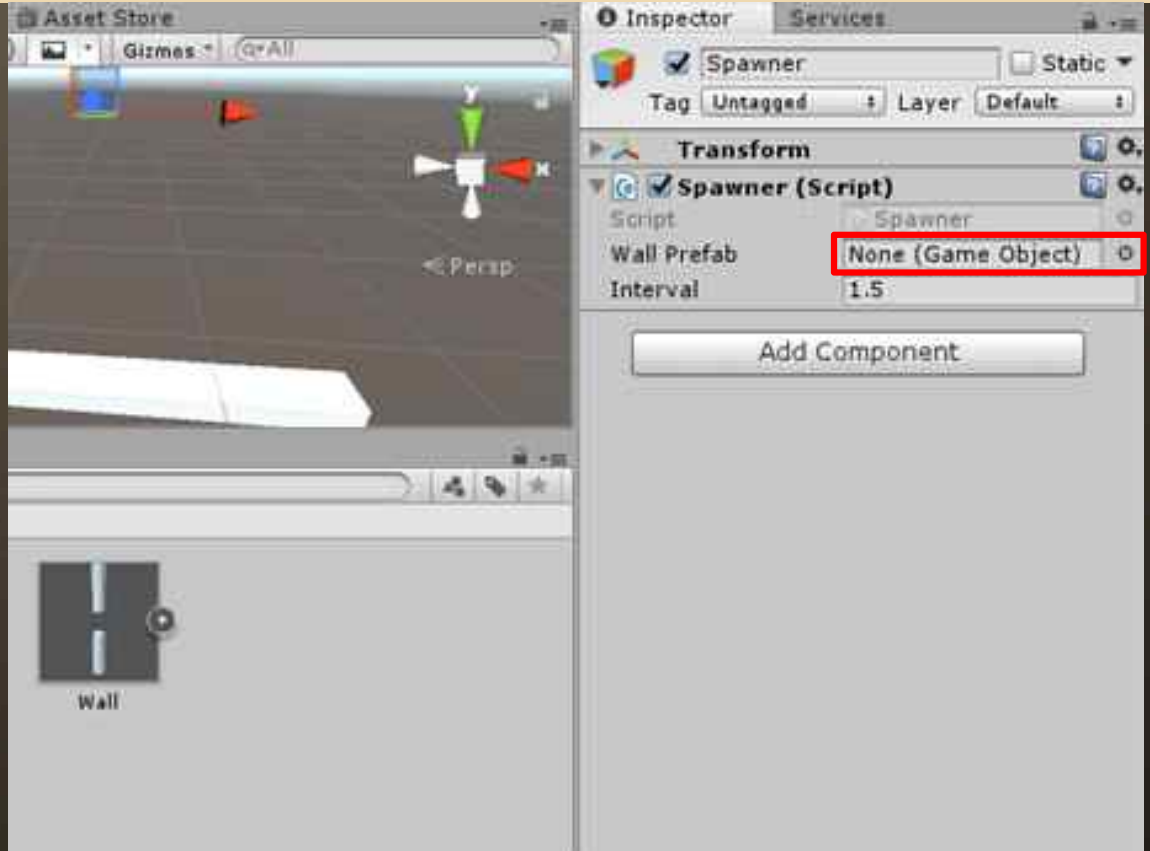

Hierarchy
Create
game
Main Camera
Directional Light
Player
Floor
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Spawner
Wall
Wall


Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Spawner (Script)
Script
Spawner
Wall Prefab
None (Game Object)
Interval
1.5

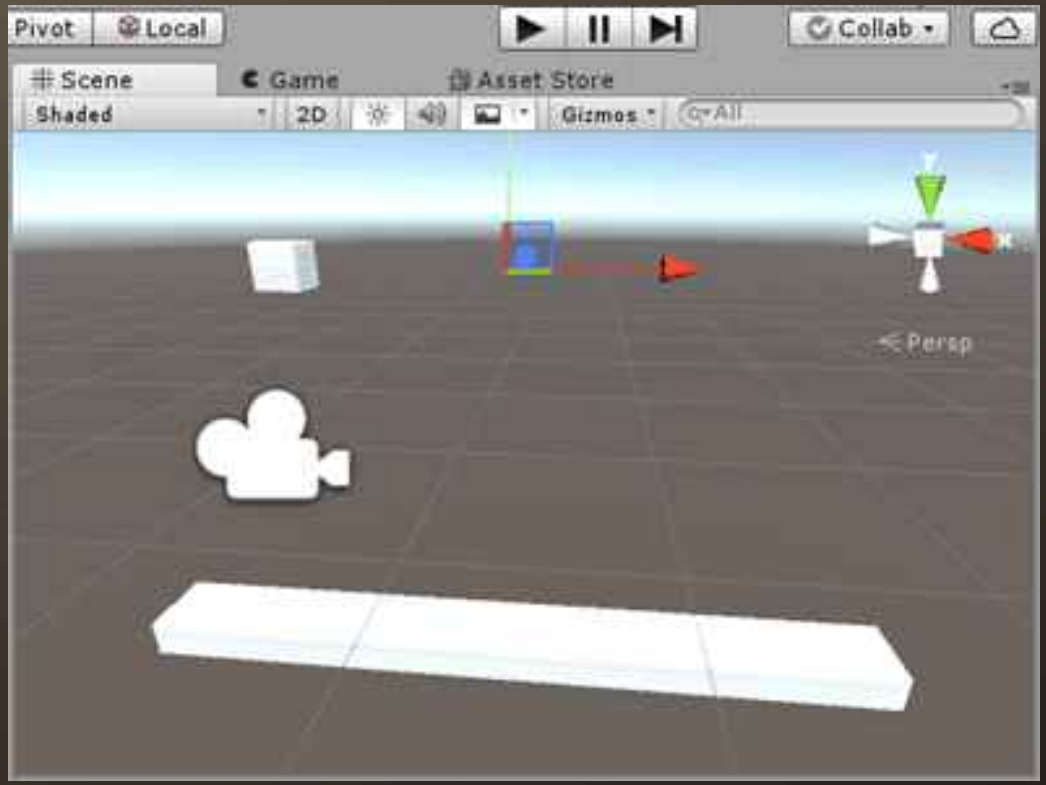
프리팹 연결 (선택 방식)


Asset Store
Gizmos
Persp
Wall
Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Spawner (Script)
Script
Spawner
Wall Prefab
None (Game Object)
Interval
1.5
Add Component

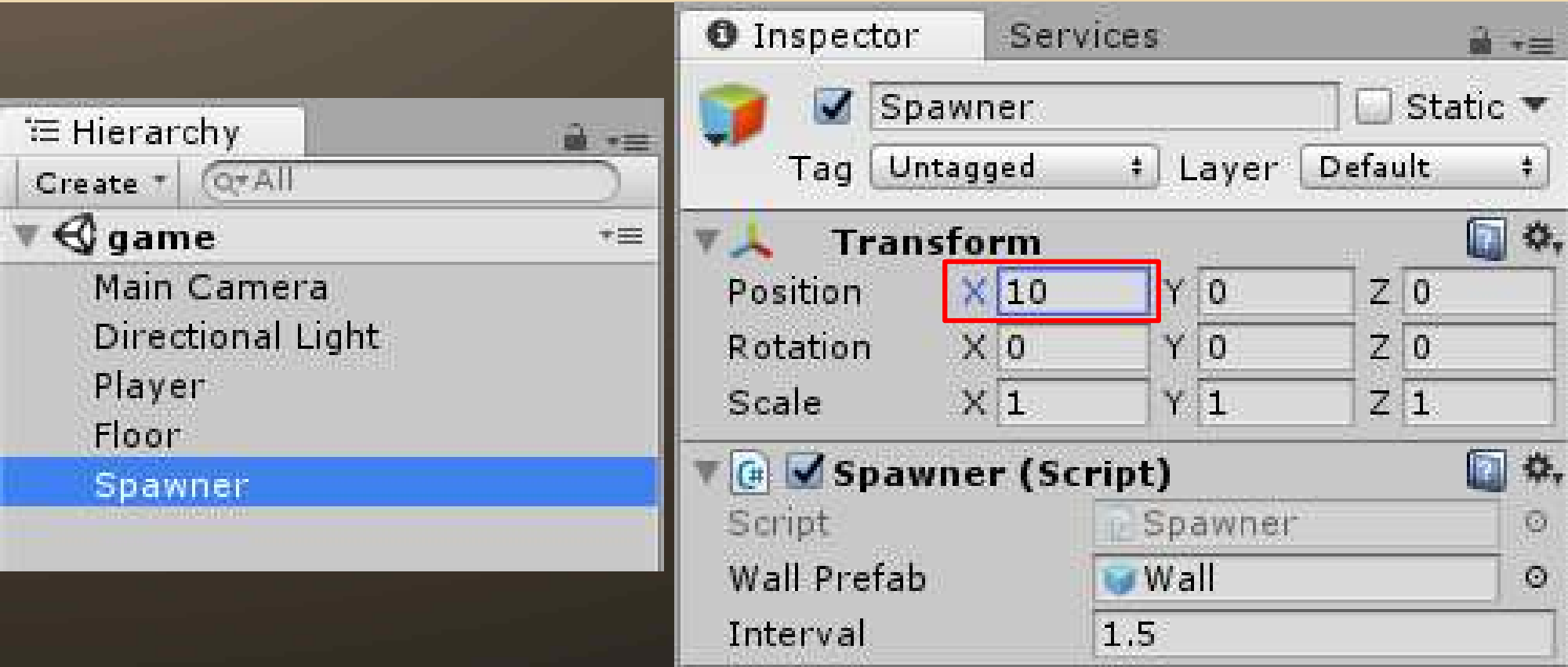
프리팹 연결 (드래그&드롭 방식)



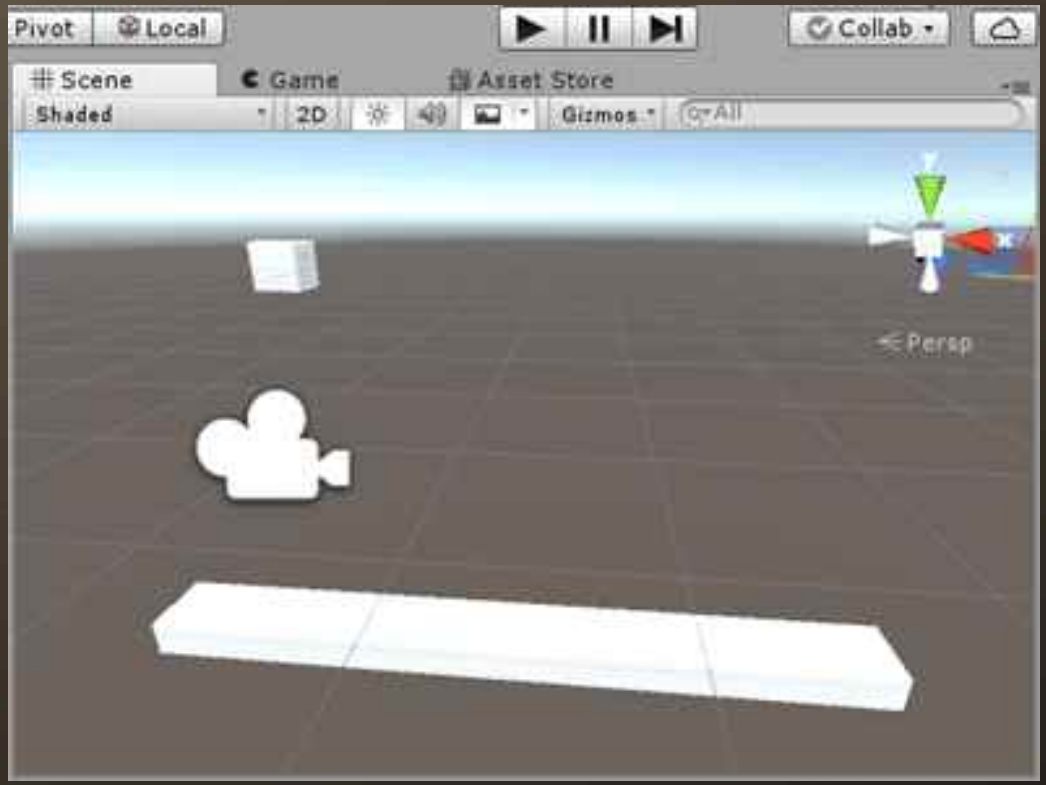
플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp

스폰 위치 조정


Hierarchy
Create
All
game
Main Camera
Directional Light
Player
Floor
Spawner
Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Position
X 10
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Spawner (Script)
Script
Spawner
Wall Prefab
Wall
Interval
1.5

플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp

벽의 높이 조정

Spawner.cs | MiniGame | Spawner | Start()

```
public class Spawner : MonoBehaviour {

    public GameObject wallPrefab;
    public float interval = 1.5f;
    public float range = 3;
    float term;

    // Use this for initialization
    void Start () {
        term = interval;
    }
```

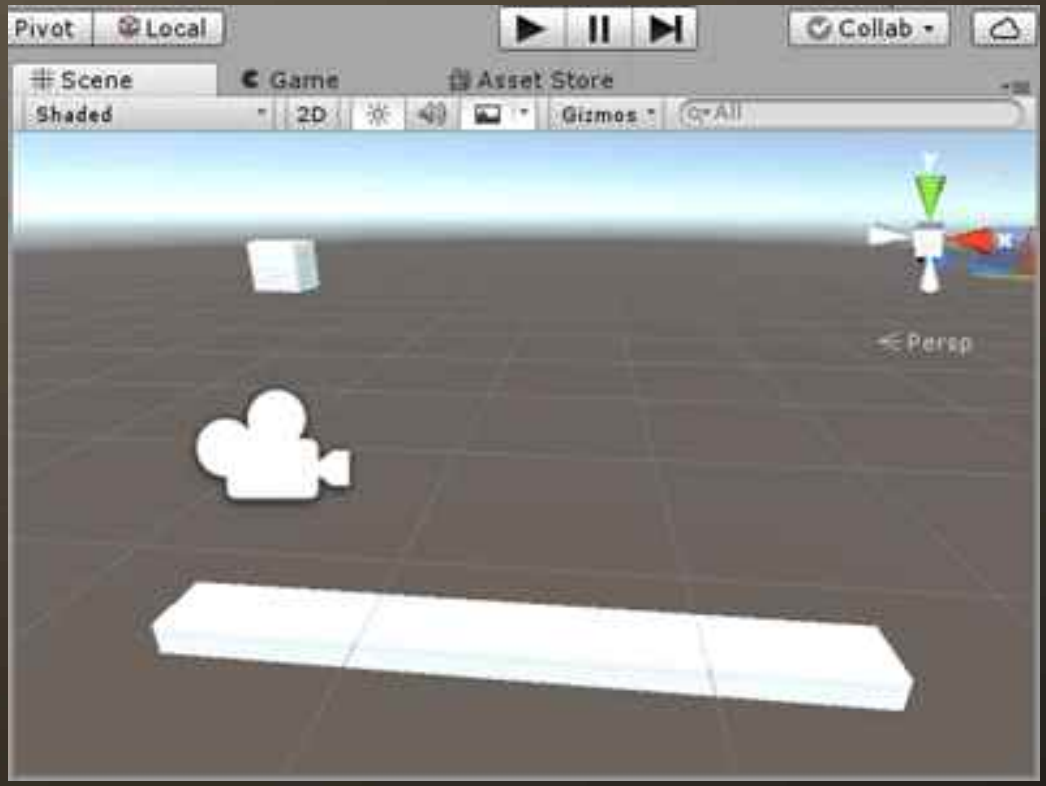
벽의 높이 조정

Spawner.cs

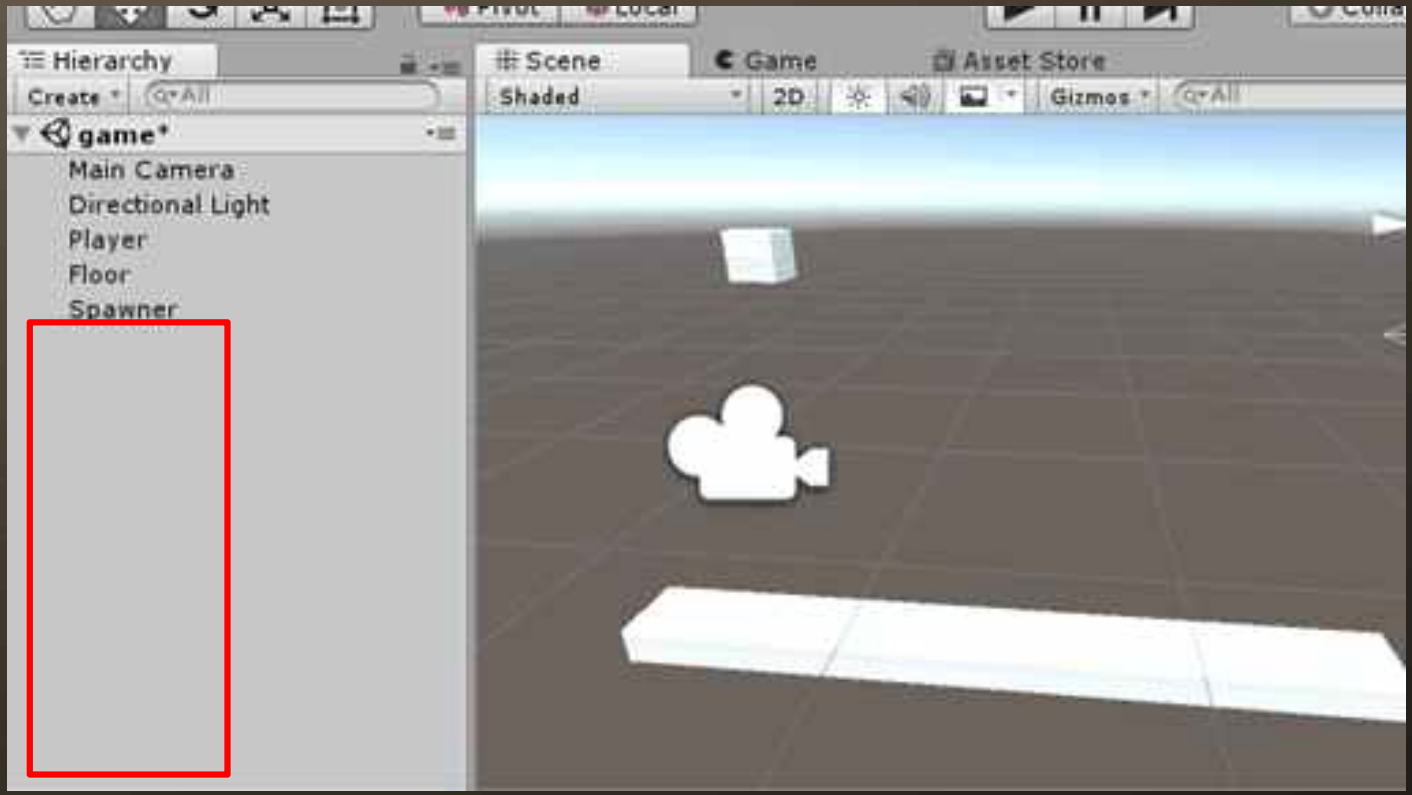
MiniGame | Spawner | Update()

```
    // Update is called once per frame
    void Update () {
        term += Time.deltaTime;
        if (term >= interval)
        {
            Vector3 pos = transform.position;
            pos.y += Random.Range(-range, range);
            Instantiate(wallPrefab, pos,
transform.rotation);
            term -= interval;
        }
    }
}
```

플레이


Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp

플레이


Hierarchy
Create
game*
Main Camera
Directional Light
Player
Floor
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos

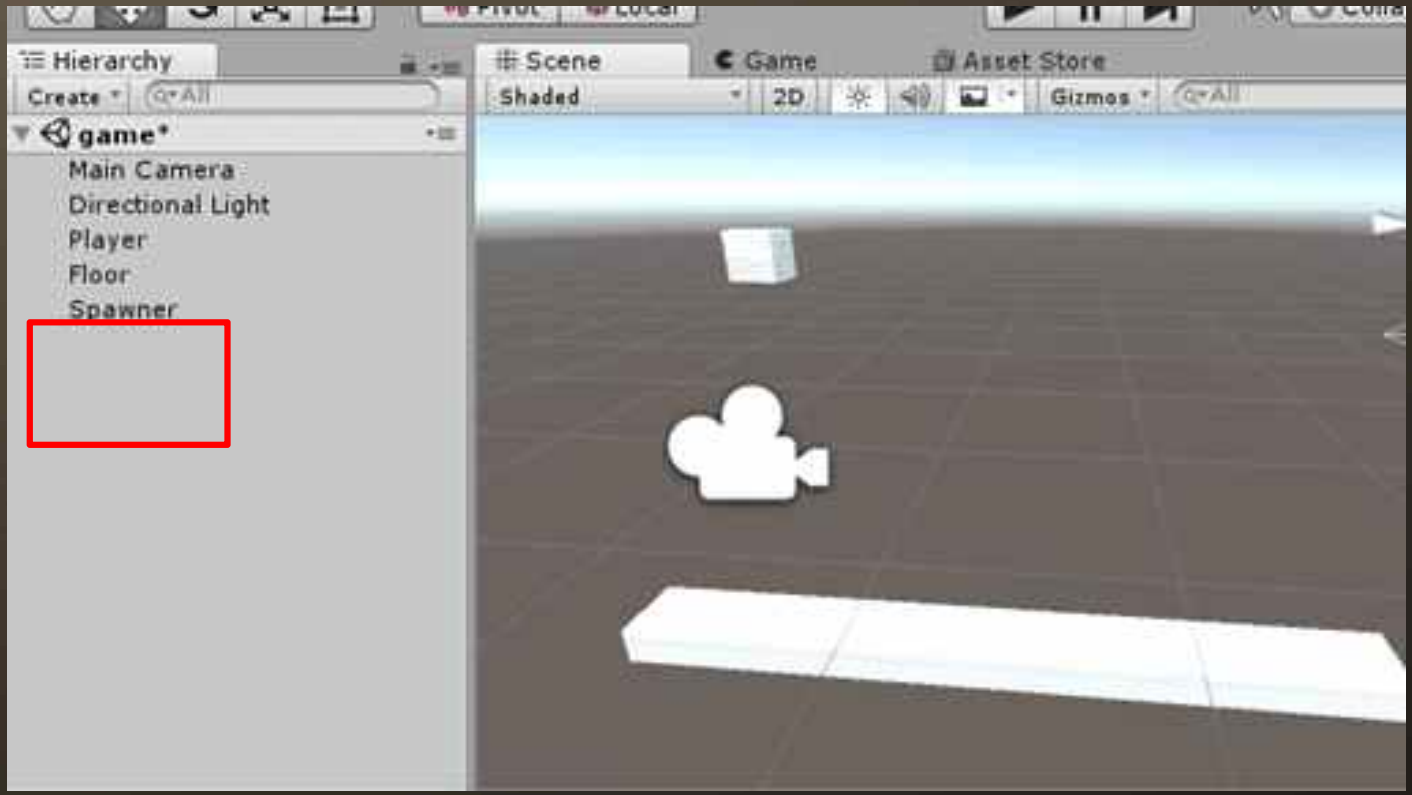
벽이 자동으로 사라지게

Wall.cs

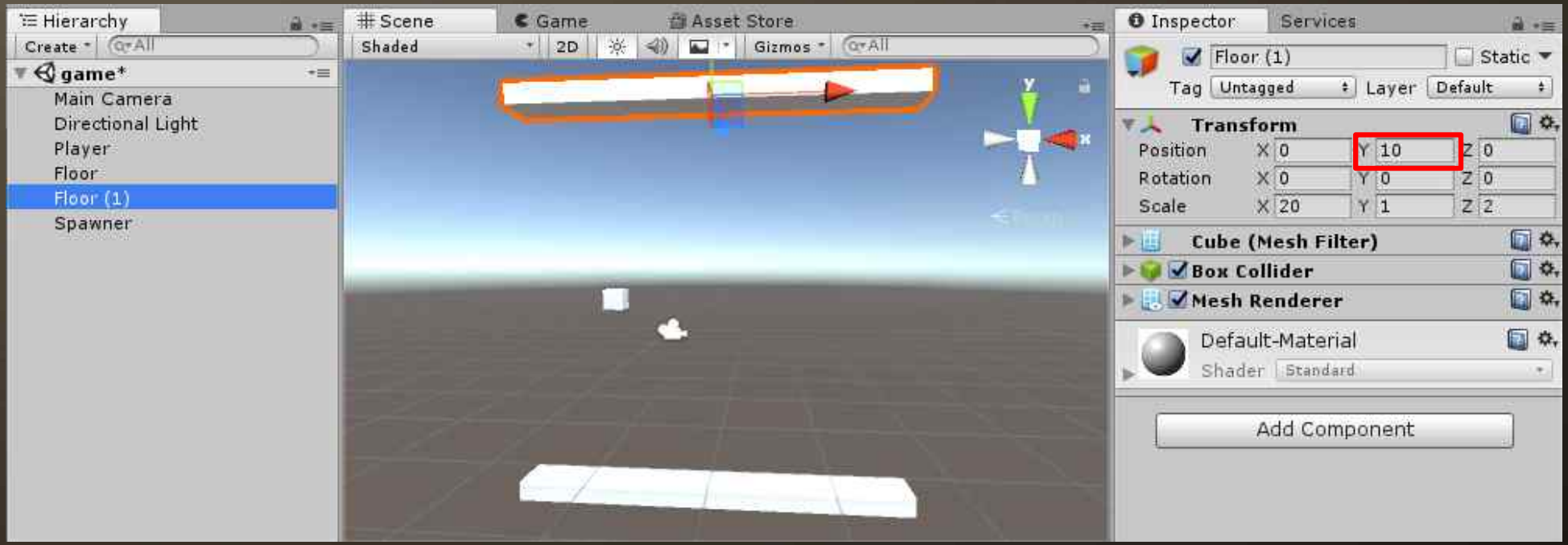
MiniGame | Wall | Update()

```
public class Wall : MonoBehaviour {

        public float speed = -5;

        // Use this for initialization
        void Start () {

        }

        // Update is called once per frame
        void Update () {
                transform.Translate(speed * Time.deltaTime, 0, 0);
                if (transform.position.x < -10)
                        Destroy(gameObject);
        }
}
```

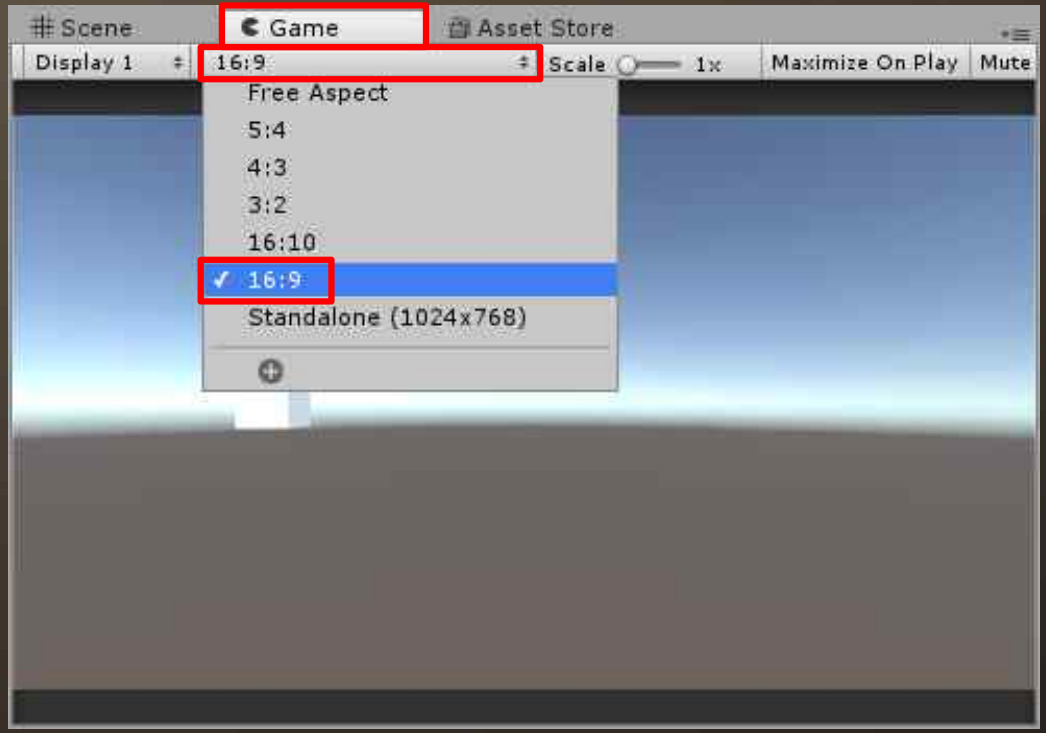
플레이


Hierarchy
Create
game*
Main Camera
Directional Light
Player
Floor
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos

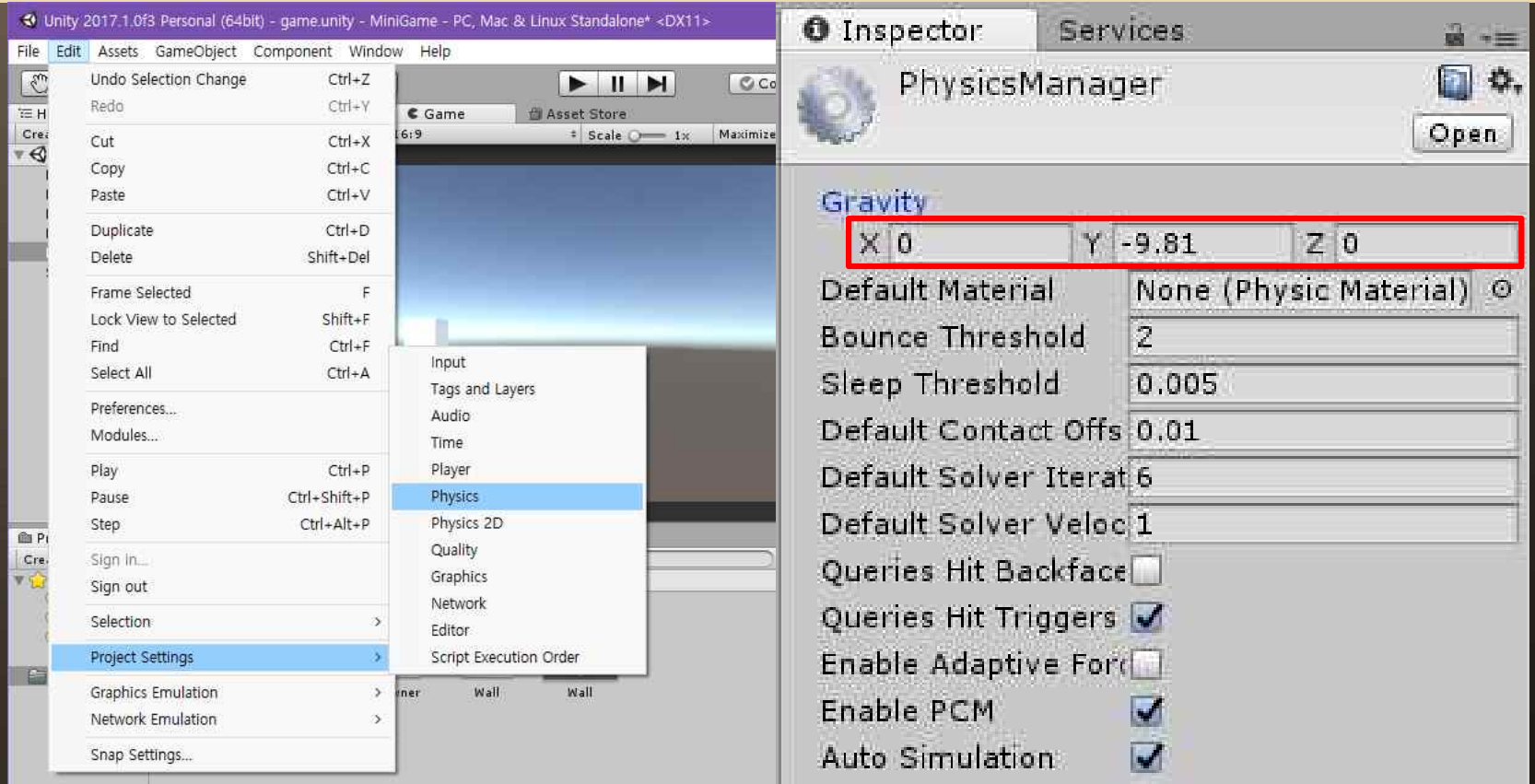
천정 추가하기


Hierarchy
Create
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos
Inspector
Services
Floor (1)
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Box Collider
Mesh Renderer
Default-Material
Shader
Standard
Add Component

화면 비율 고정하기


Scene
Game
Asset Store
Display 1
16:9
Scale
1x
Maximize On Play
Mute
Free Aspect
5:4
4:3
3:2
16:10
16:9
Standalone (1024x768)

중력 변경하기


Unity 2017.1.0f3 Personal (64bit) - game.unity - MiniGame - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Window Help
Undo Selection Change Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Duplicate Ctrl+D
Delete Shift+Del
Frame Selected F
Lock View to Selected Shift+F
Find Ctrl+F
Select All Ctrl+A
Preferences...
Modules...
Play Ctrl+P
Pause Ctrl+Shift+P
Step Ctrl+Alt+P
Sign in...
Sign out
Selection
Project Settings
Graphics Emulation
Network Emulation
Snap Settings...
Game
Asset Store
Scale
Maximize
Input
Tags and Layers
Audio
Time
Player
Physics
Physics 2D
Quality
Graphics
Network
Editor
Script Execution Order
Wall
Wall
Inspector
Services
PhysicsManager
Open
Gravity
X 0
Y -9.81
Z 0
Default Material None (Physic Material)
Bounce Threshold 2
Sleep Threshold 0.005
Default Contact Offs 0.01
Default Solver Iterat 6
Default Solver Veloc 1
Queries Hit Backface
Queries Hit Triggers
Enable Adaptive For
Enable PCM
Auto Simulation

그 밖의 여러가지 변경해 보기

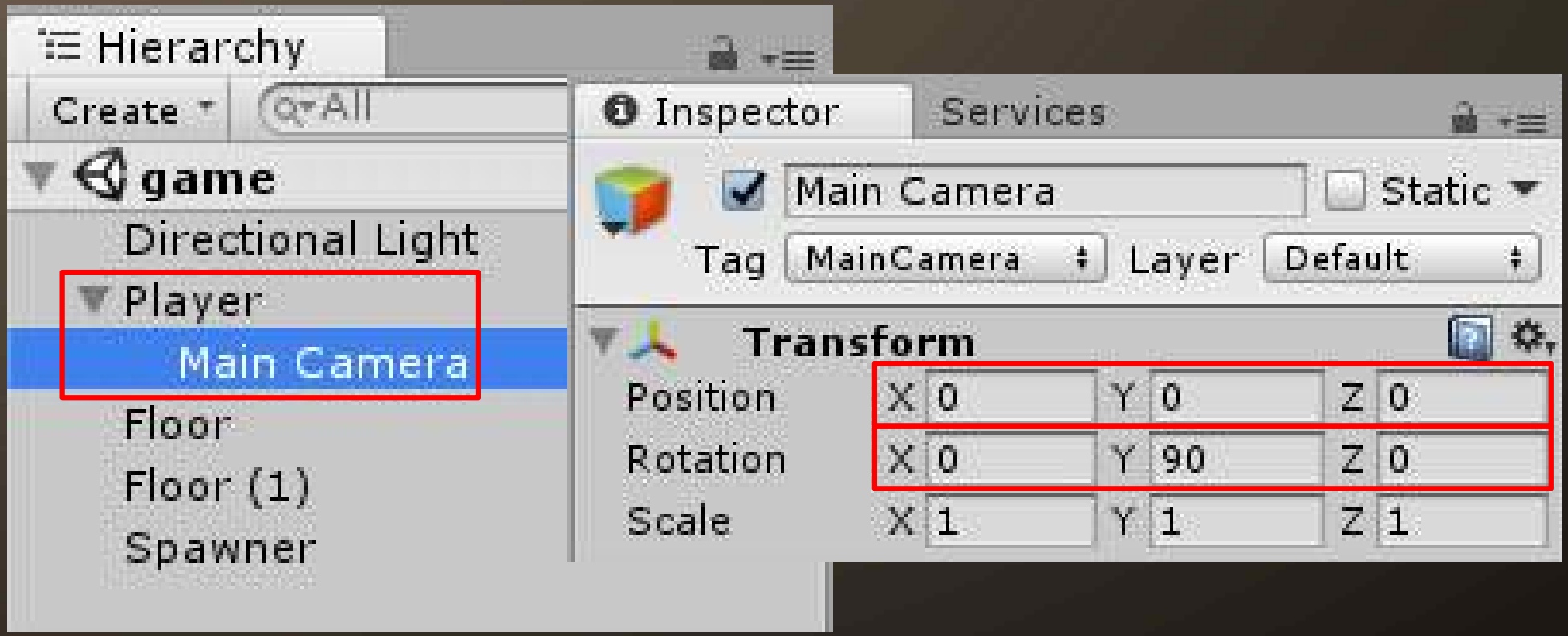
- 중력
- 점프력 (Player의 Jump Power)
- 벽의 속도 (Wall 프리팹의 Speed)
- 벽의 간격 (Spawner의 Interval)
- 벽의 높이 차이 (Spawner의 Range)
- 플레이어의 모델, 장애물의 모델

다양한 시도

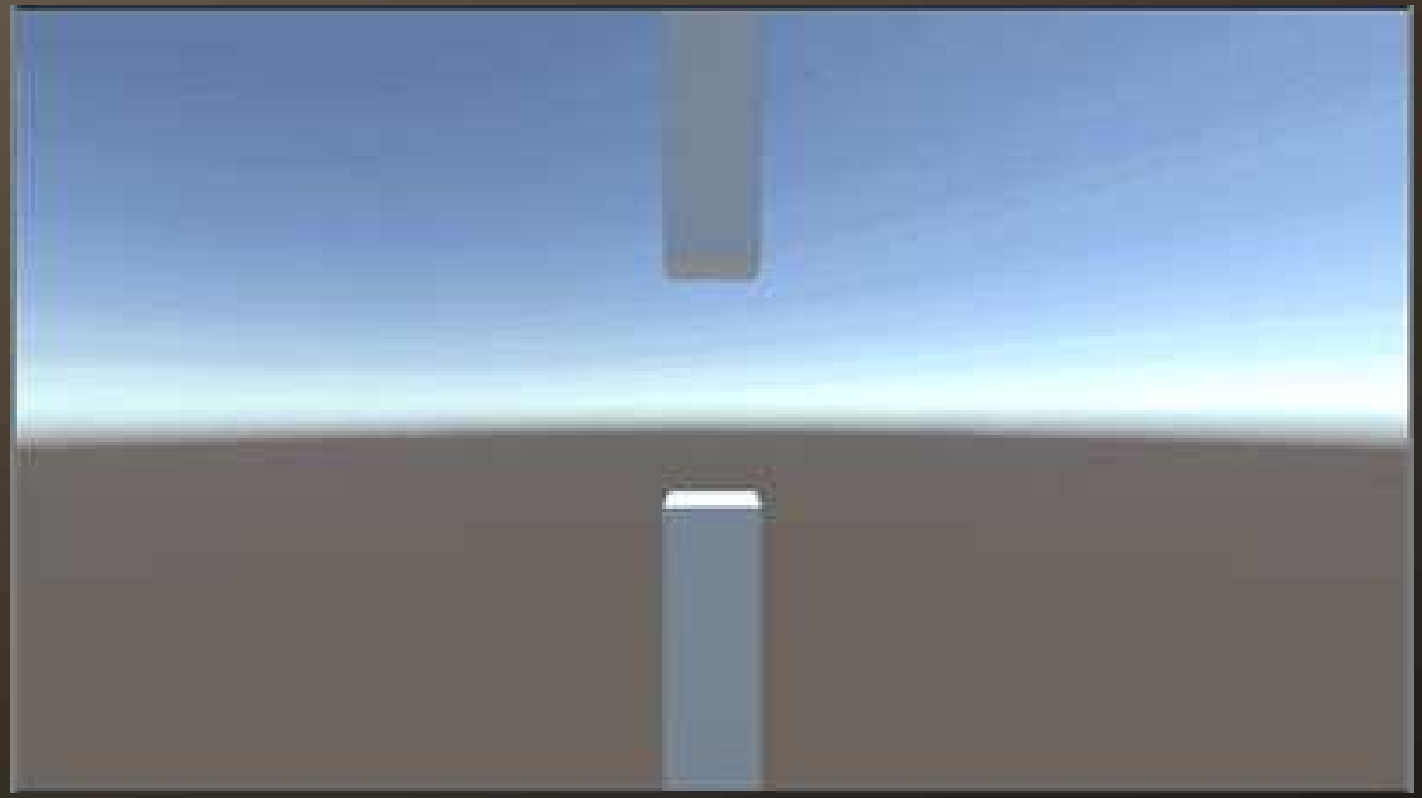
Variations

Try 1
1인칭 시점

시도1: 1인칭 시점


Hierarchy
Create
All
game
Directional Light
Player
Main Camera
Floor
Floor (1)
Spawner
Inspector
Services
Main Camera
Static
Tag
MainCamera
Layer
Default
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 90
Z 0
Scale
X 1
Y 1
Z 1

시도1: 1인칭 시점



Try 2
낮은 높이에서 점프 부스터

시도2: 낮은 높이에서 점프 부스터

Player.cs

MiniGame | Player | Start()

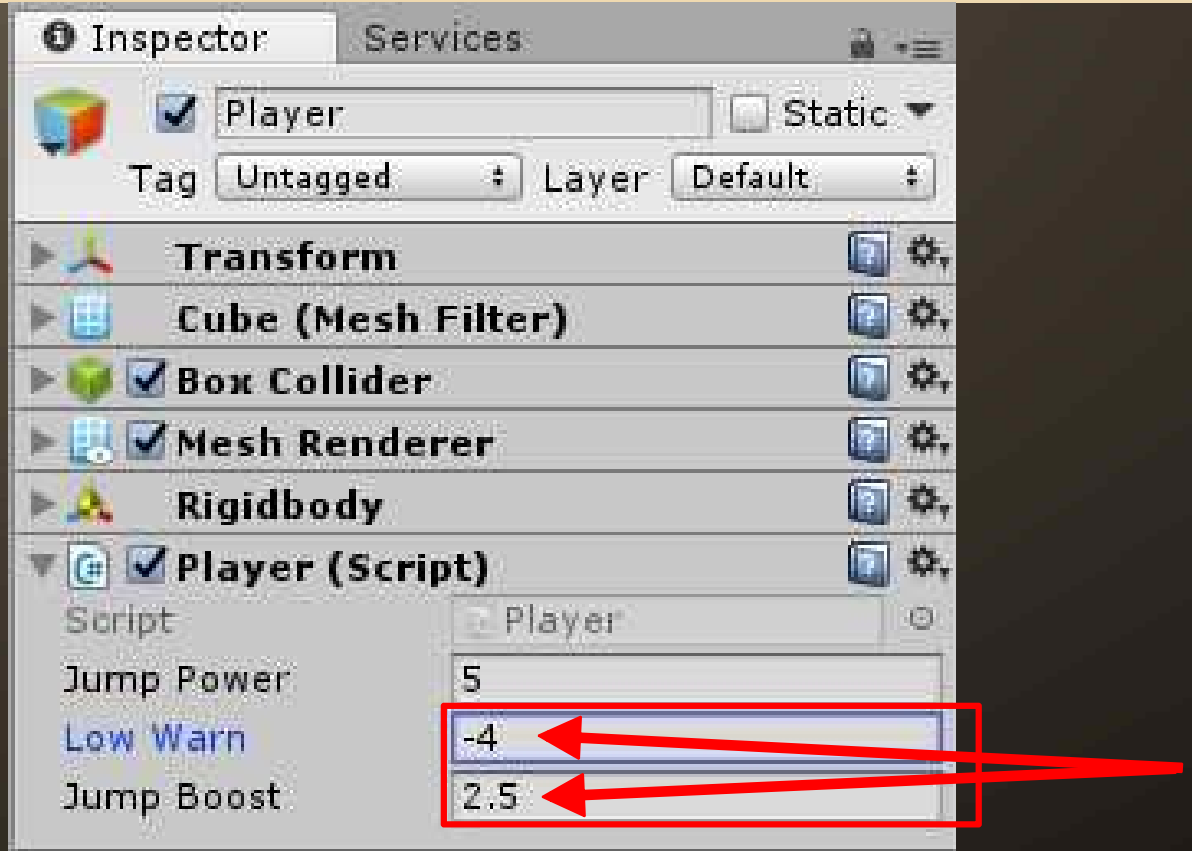
```
public class Player : MonoBehaviour {

    public float jumpPower = 5;
    public float lowWarn = -4;
    public float jumpBoost = 2.5f;

    // Use this for initialization
    void Start () {

    }
```

시도2: 낮은 높이에서 점프 부스터

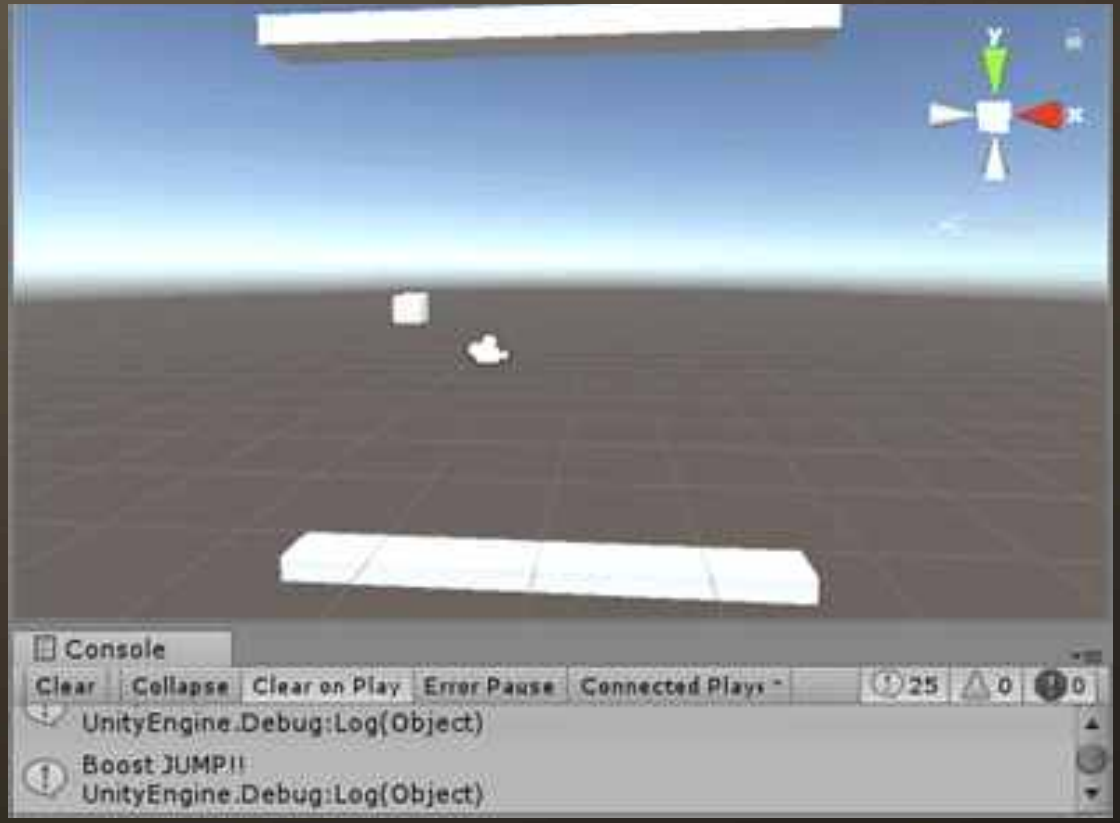
Player.cs | MiniGame | Player | Update()

```
        // Update is called once per frame
        void Update () {
                if (Input.GetButtonDown("Jump"))
                {
                        if (transform.position.y < lowWarn)
                        {
                                GetComponent<Rigidbody>().velocity =
                                        new
Vector3(0, jumpPower * jumpBoost, 0);
                                Debug.Log("Boost JUMP!!");
                        }
                        else
                        {
                                GetComponent<Rigidbody>().velocity =
                                        new
Vector3(0, jumpPower, 0);
                                Debug.Log("Jump.");
                        }
```

시도2: 낮은 높이에서 점프 부스터



시도2: 낮은 높이에서 점프 부스터



Try 3
점점 앞으로 가는 플레이어

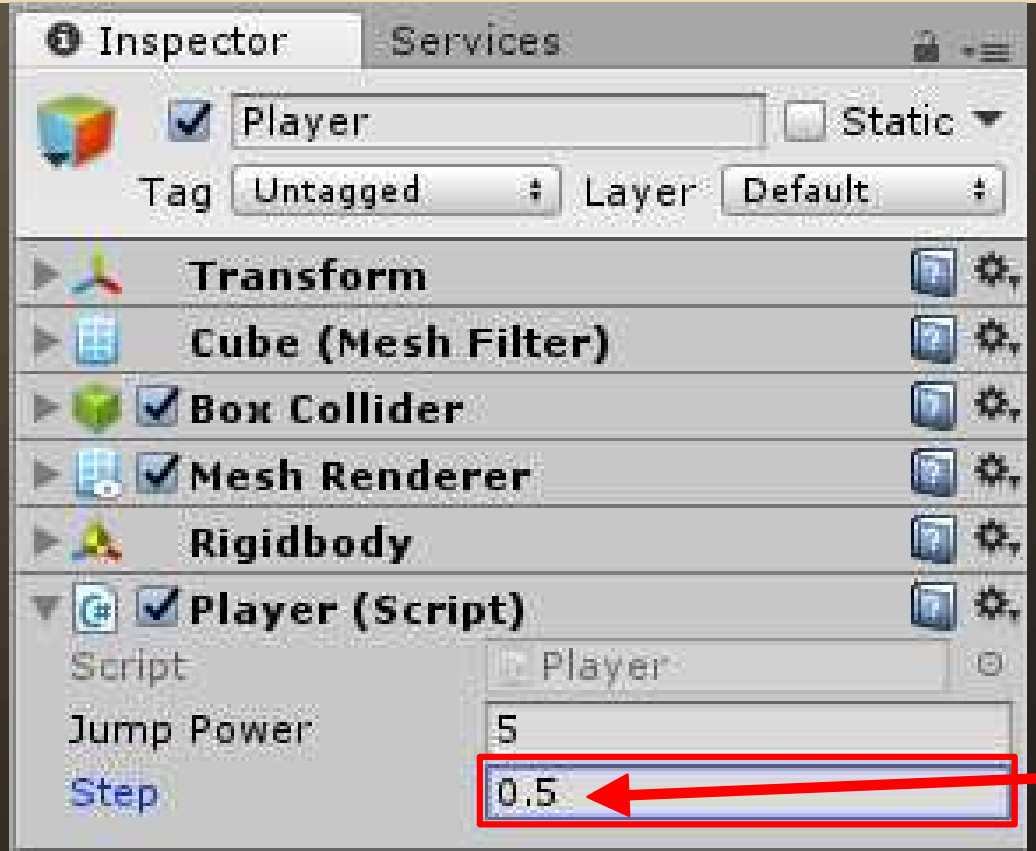
시도3: 점점 앞으로 가는 플레이어

Player.cs

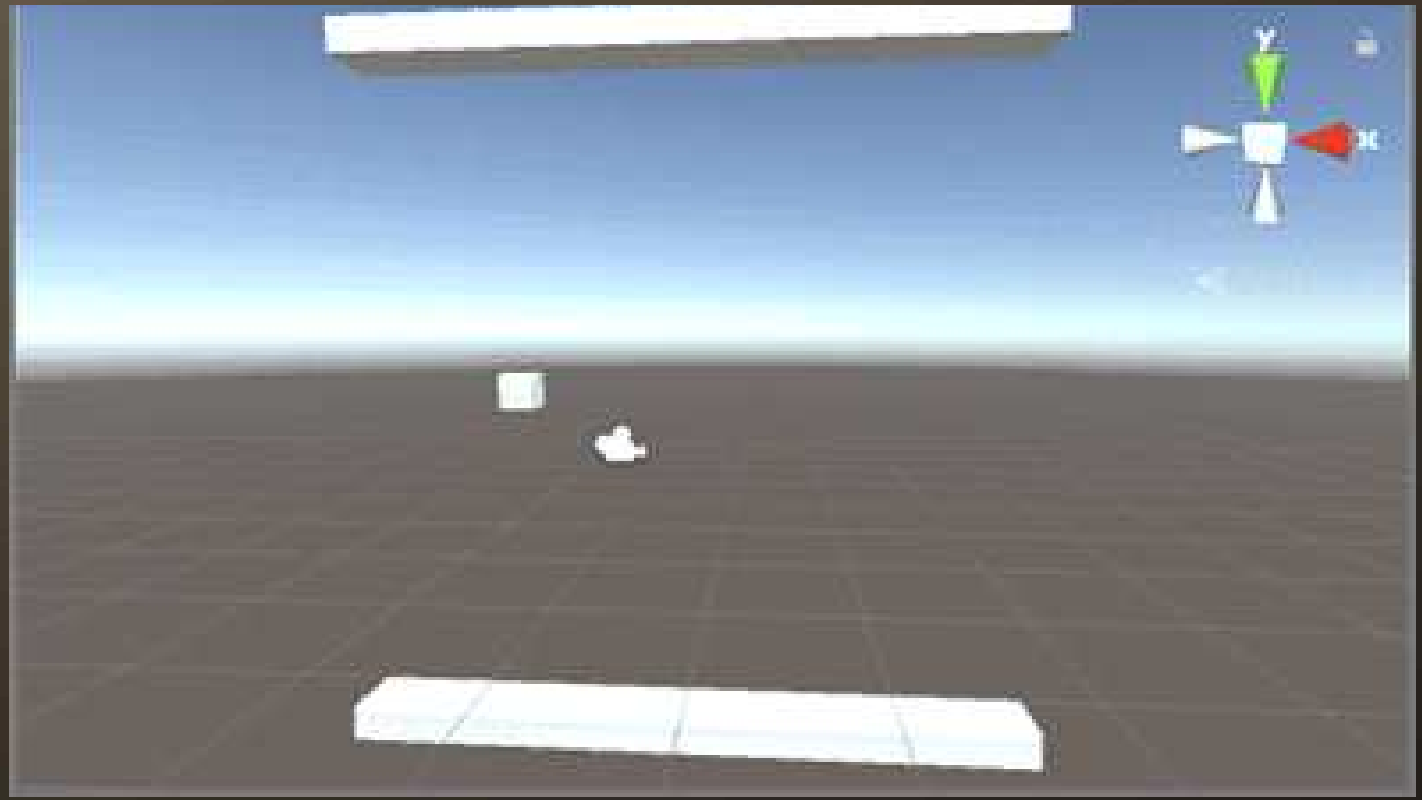
MiniGame | Player | Update()

```
public class Player : MonoBehaviour {

        public float jumpPower = 5;
        public float step = 0.5f;

        // Use this for initialization
        void Start () {

        }

        // Update is called once per frame
        void Update () {
                transform.position += new Vector3(step * Time.deltaTime, 0, 0);
                if (Input.GetButtonDown("Jump"))
                                GetComponent<Rigidbody>().velocity = new Vector3(0,
jumpPower, 0);
        }
```

시도3: 점점 앞으로 가는 플레이어



시도3: 점점 앞으로 가는 플레이어



Try 4
점점 키가 크는 플레이어

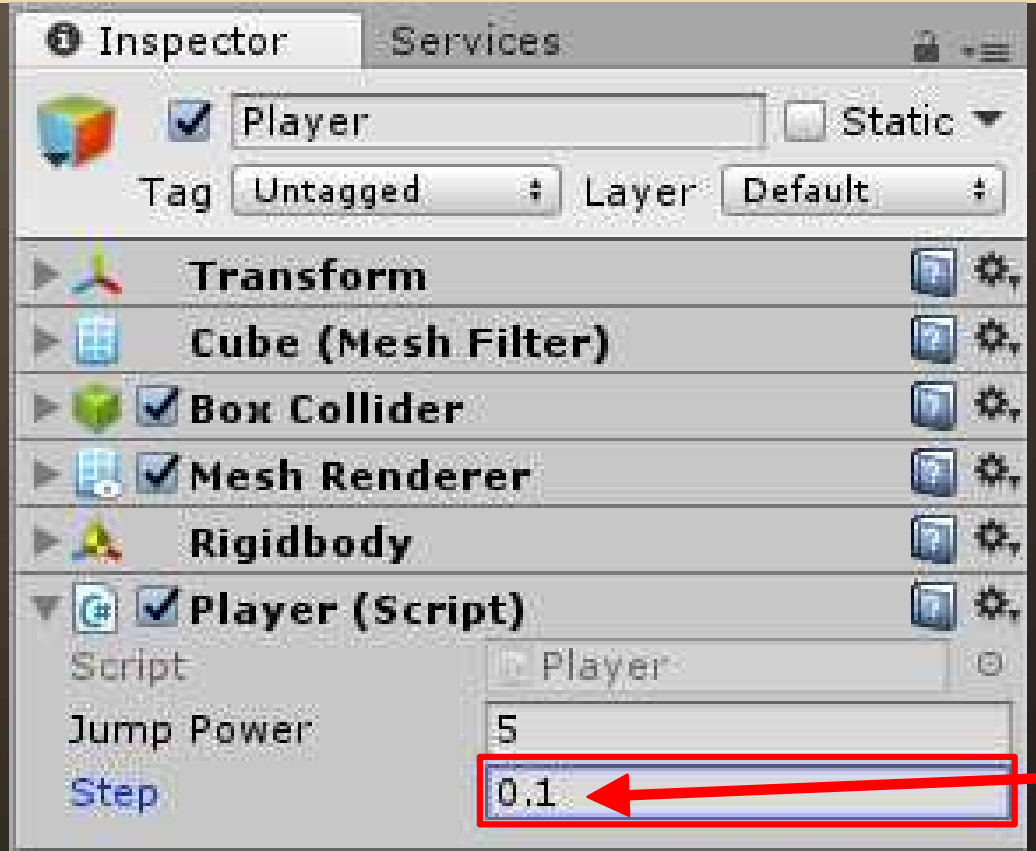
시도4: 점점 키가 크는 플레이어

Player.cs

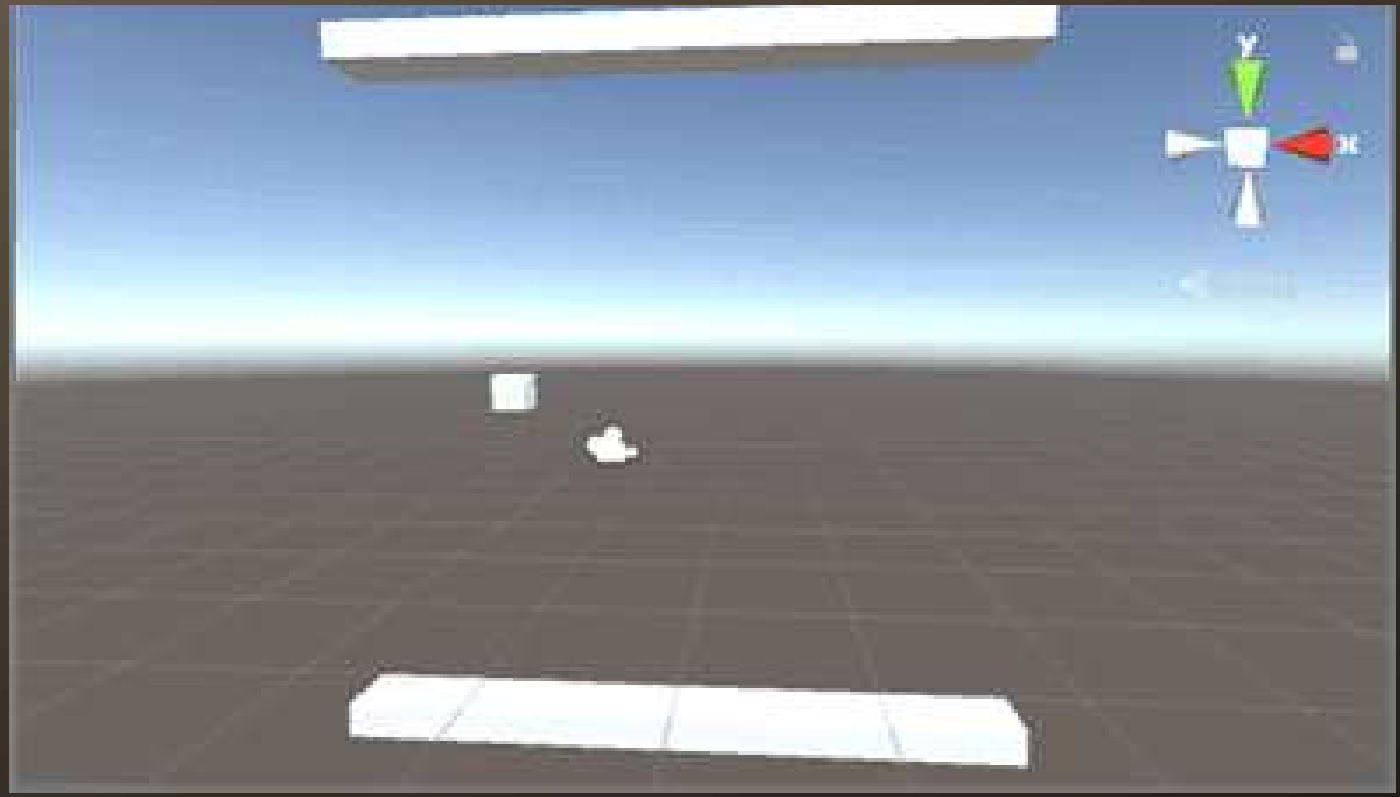
MiniGame | Player | Update()

```
public class Player : MonoBehaviour {

        public float jumpPower = 5;
        public float step = 0.1f;

        // Use this for initialization
        void Start () {

        }

        // Update is called once per frame
        void Update () {
                transform.localScale += new Vector3(0, step * Time.deltaTime, 0);
                if (Input.GetButtonDown("Jump"))
                                GetComponent<Rigidbody>().velocity = new Vector3(0,
jumpPower, 0);
        }
```

시도4: 점점 키가 크는 플레이어



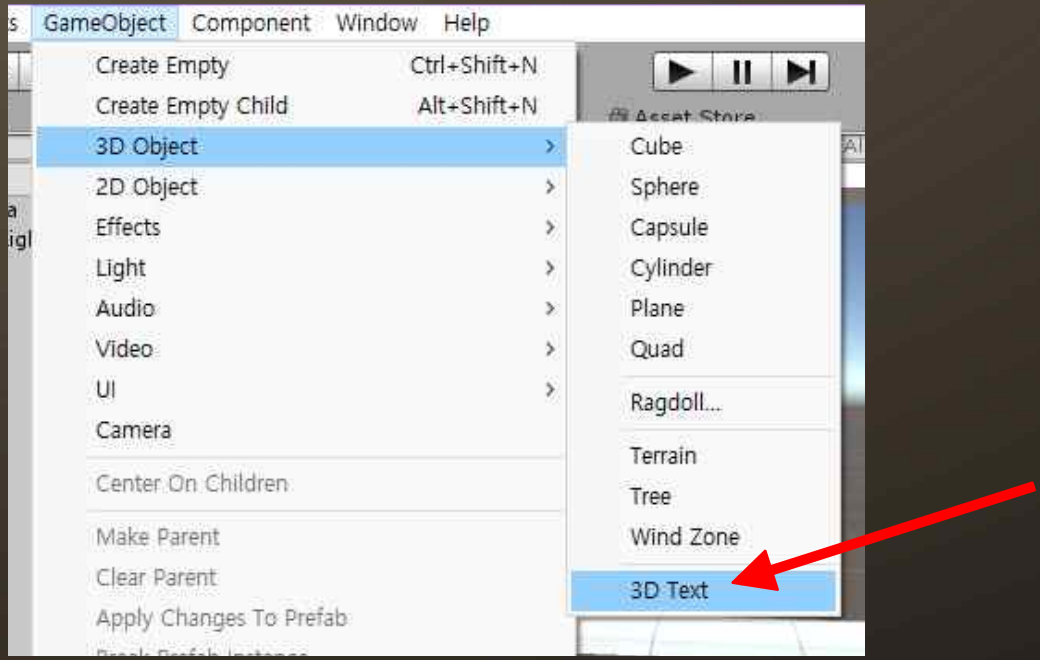
시도4: 점점 키가 크는 플레이어



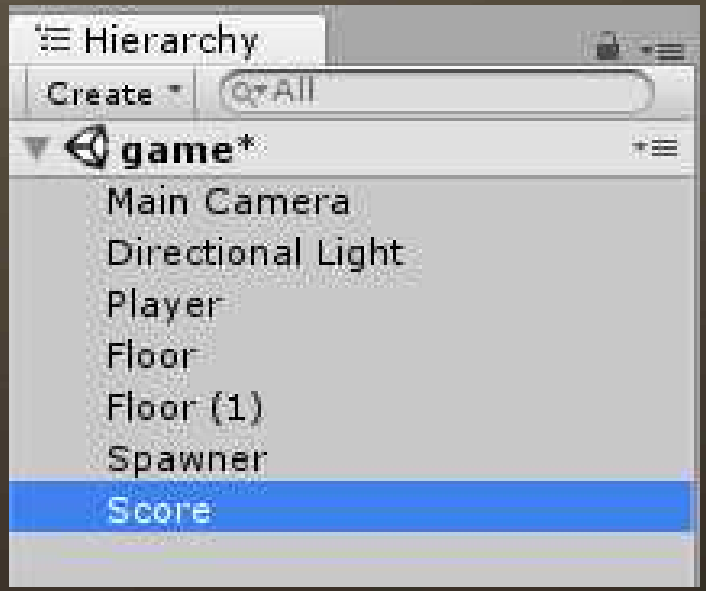
Try 5
점수 카운팅

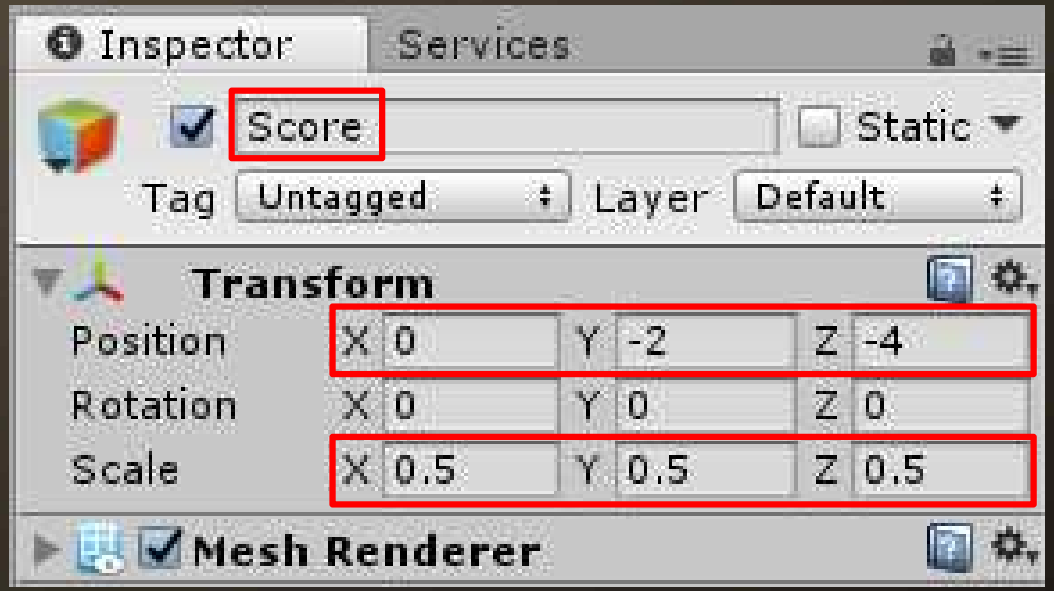
시도5: 점수 카운팅

. GameObject > 3D Object > 3D Text

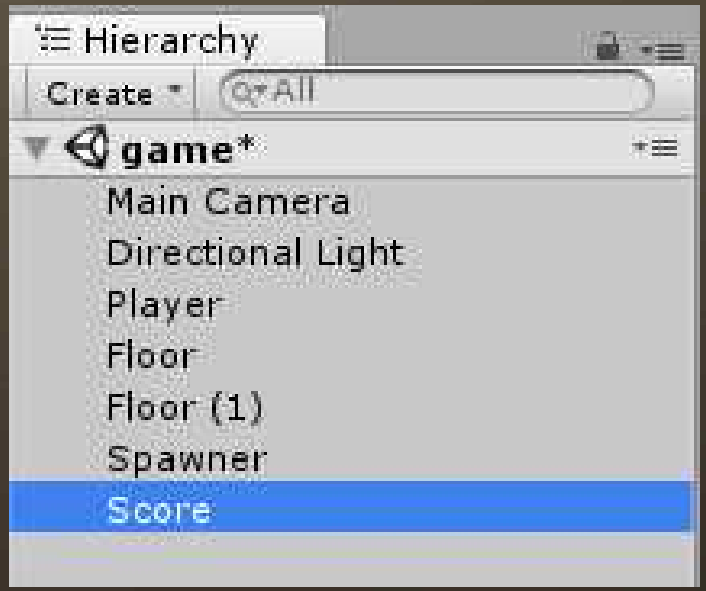


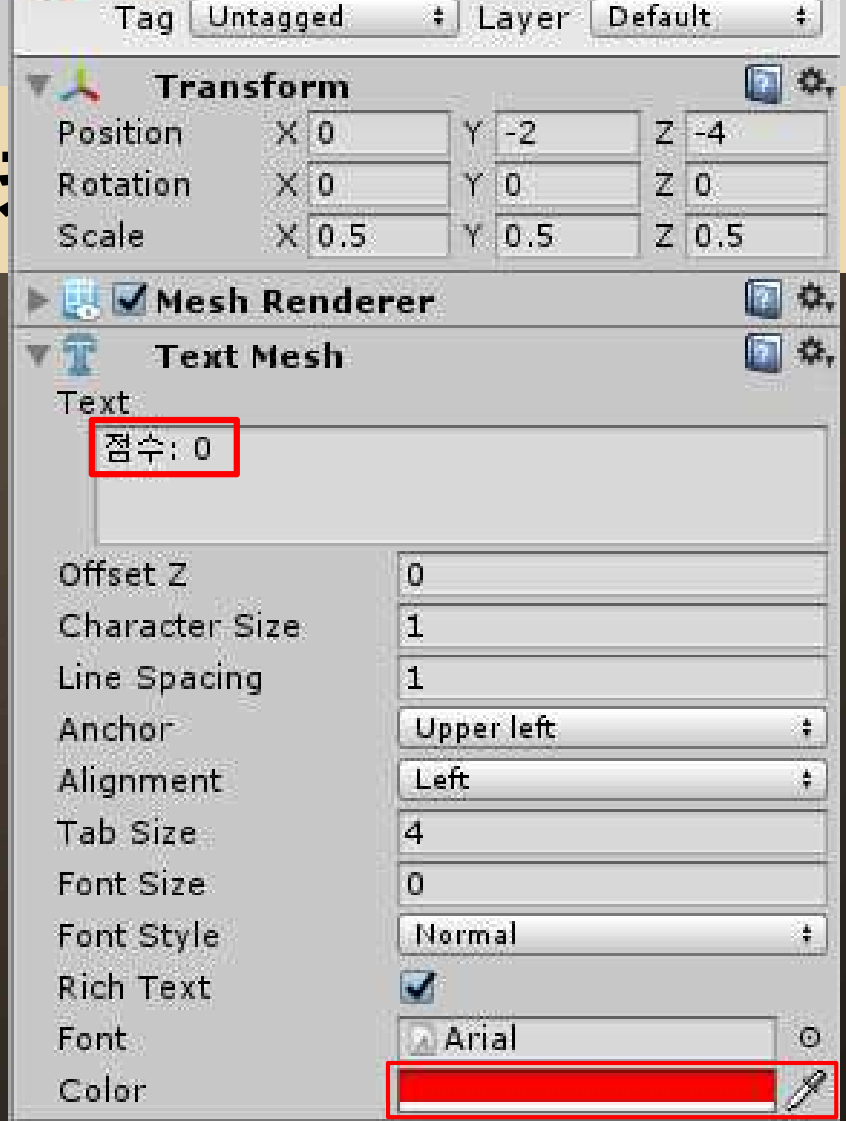
시도5: 점수 카운팅


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score


Inspector
Services
Score
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y -2
Z -4
Rotation
X 0
Y 0
Z 0
Scale
X 0.5
Y 0.5
Z 0.5
Mesh Renderer

시도5: 점수


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score


Tag
Untagged
Layer
Default
Transform
Position
X 0
Y -2
Z -4
Rotation
X 0
Y 0
Z 0
Scale
X 0.5
Y 0.5
Z 0.5
Mesh Renderer
Text Mesh
Text
점수: 0
Offset Z
0
Character Size
1
Line Spacing
1
Anchor
Upper left
Alignment
Left
Tab Size
4
Font Size
0
Font Style
Normal
Rich Text
Font
Arial
Color

시도5: 점수 카운팅

Player.cs

MiniGame | Player | Start()

```
public class Player : MonoBehaviour {

    public float jumpPower = 5;
    TextMesh scoreOutput;
    int score = 0;

    // Use this for initialization
    void Start () {
        scoreOutput = GameObject.Find(name:
"Score").GetComponent<TextMesh>();
                    // 이름으로 게임 오브젝트를 찾고, 그 중 TextMesh
컴포넌트를 얻기
```

시도5: 점수 카운팅

Player.cs

MiniGame | Player | addScore(int s)

```
                        GetComponent<Rigidbody>().velocity =
new Vector3(0, jumpPower, 0);
        }

        private void OnCollisionEnter(Collision collision)
        {
                SceneManager.LoadScene(SceneManager.GetActiveScene().
name);
        }

        // 점수 더하기
        public void addScore(int s)
```

시도5: 점수 카운팅

Wall.cs

MiniGame | Wall | Start()

```
public class Wall : MonoBehaviour {

        public float speed = -5;
        Player player;

        // Use this for initialization
        void Start () {
                player = GameObject.Find(name:
"Player").GetComponent<Player>();
        }
```

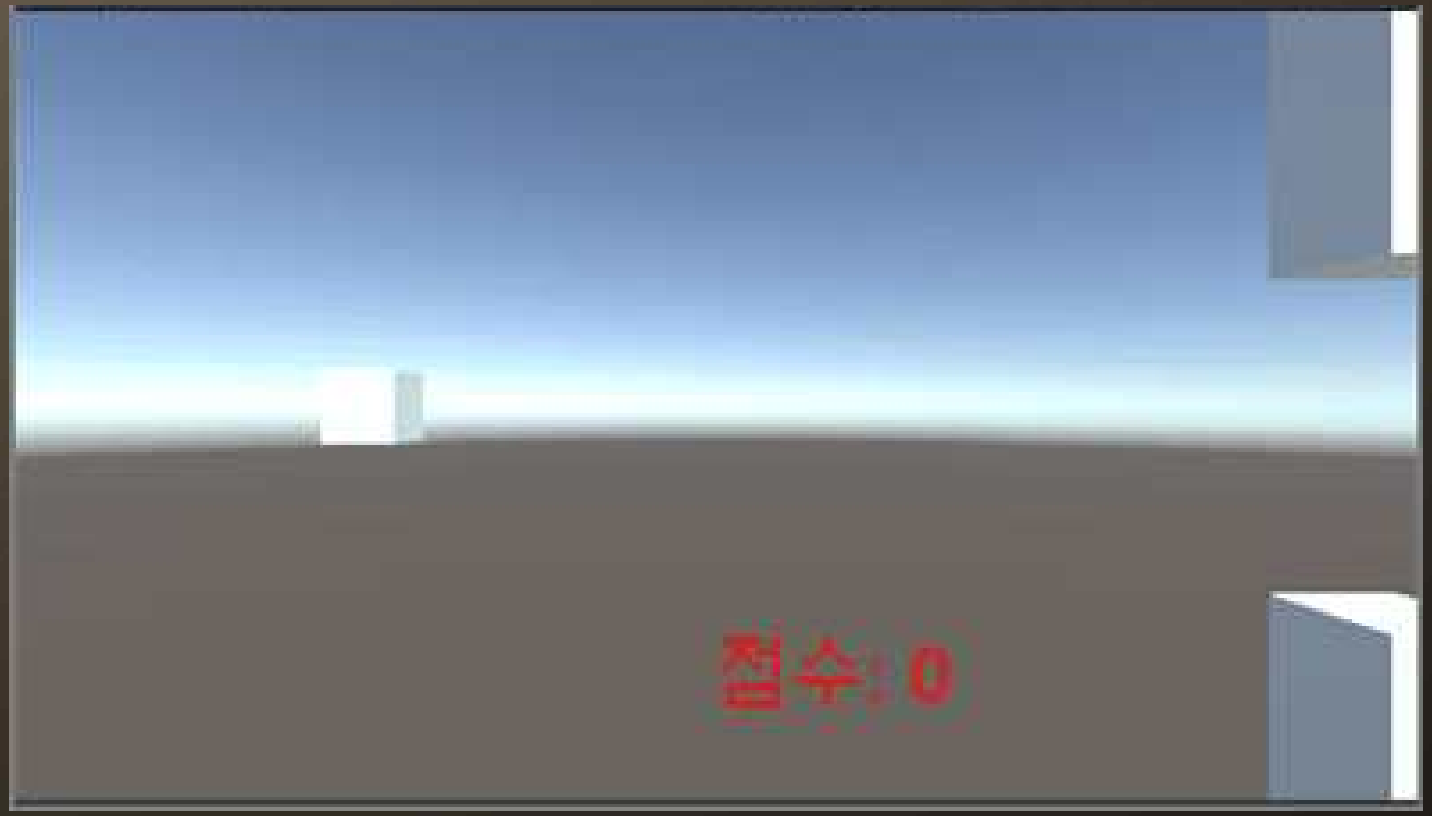
시도5: 점수 카운팅

Wall.cs

MiniGame | Wall | Update()

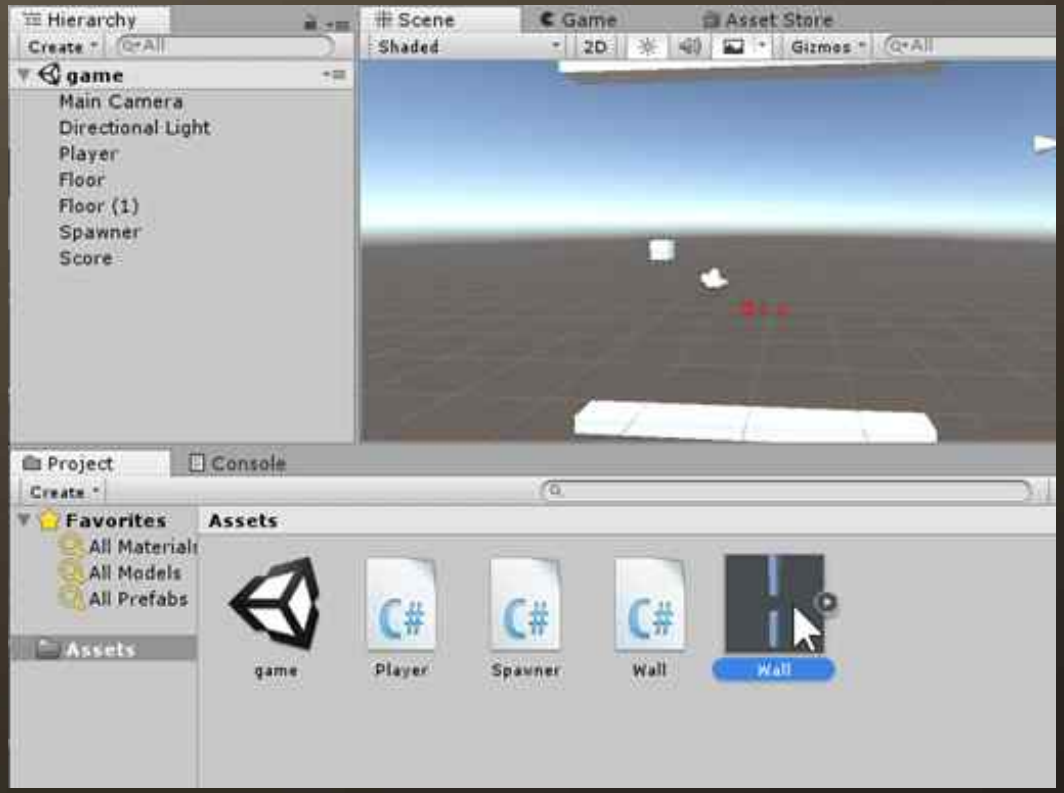
```
    // Update is called once per frame
    void Update () {
        transform.Translate(speed * Time.deltaTime, 0, 0);
        if (transform.position.x < -10)
        {
            Destroy(gameObject);
            player.addScore(1);
        }
    }
}
```

시도5: 점수 카운팅



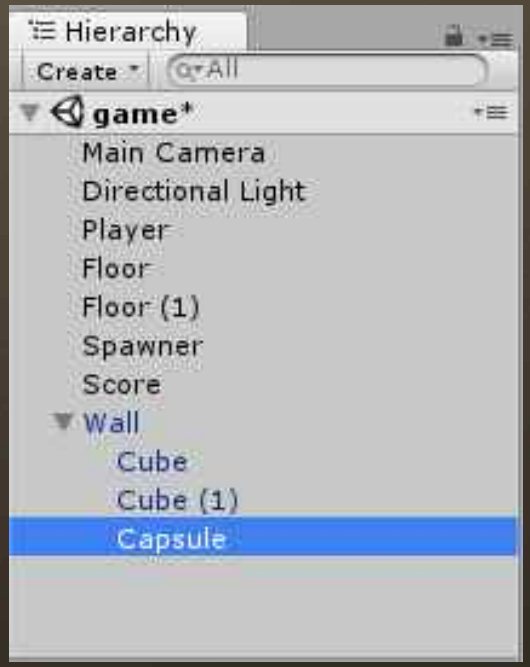
Try 6
다양한 종류의 벽

시도6: 다양한 종류의 벽

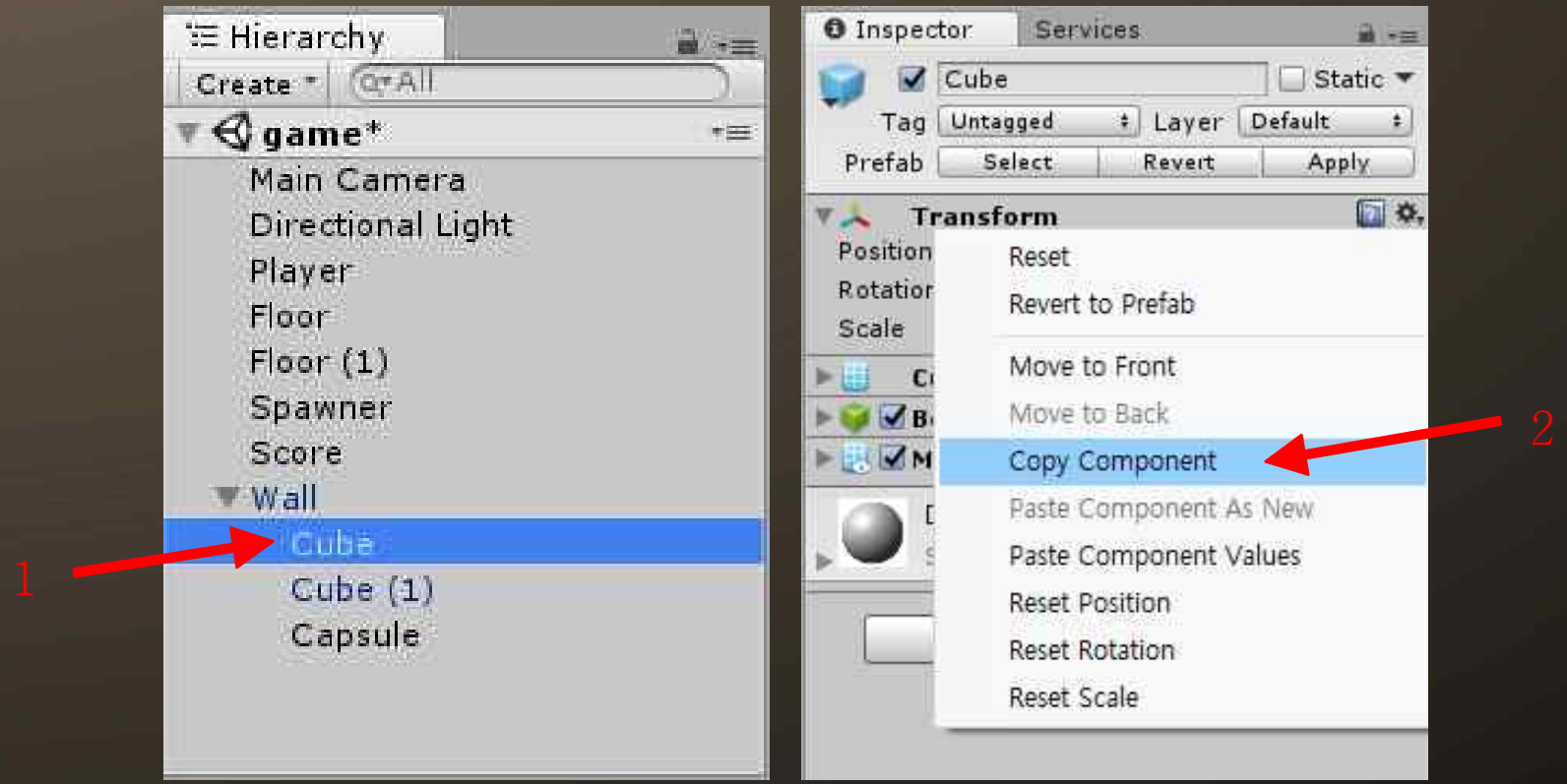

Hierarchy
Create
game
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Scene
Game
Asset Store
Shaded
2D
Gizmos
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Spawner
Wall
Wall

시도6: 다양한 종류의 벽

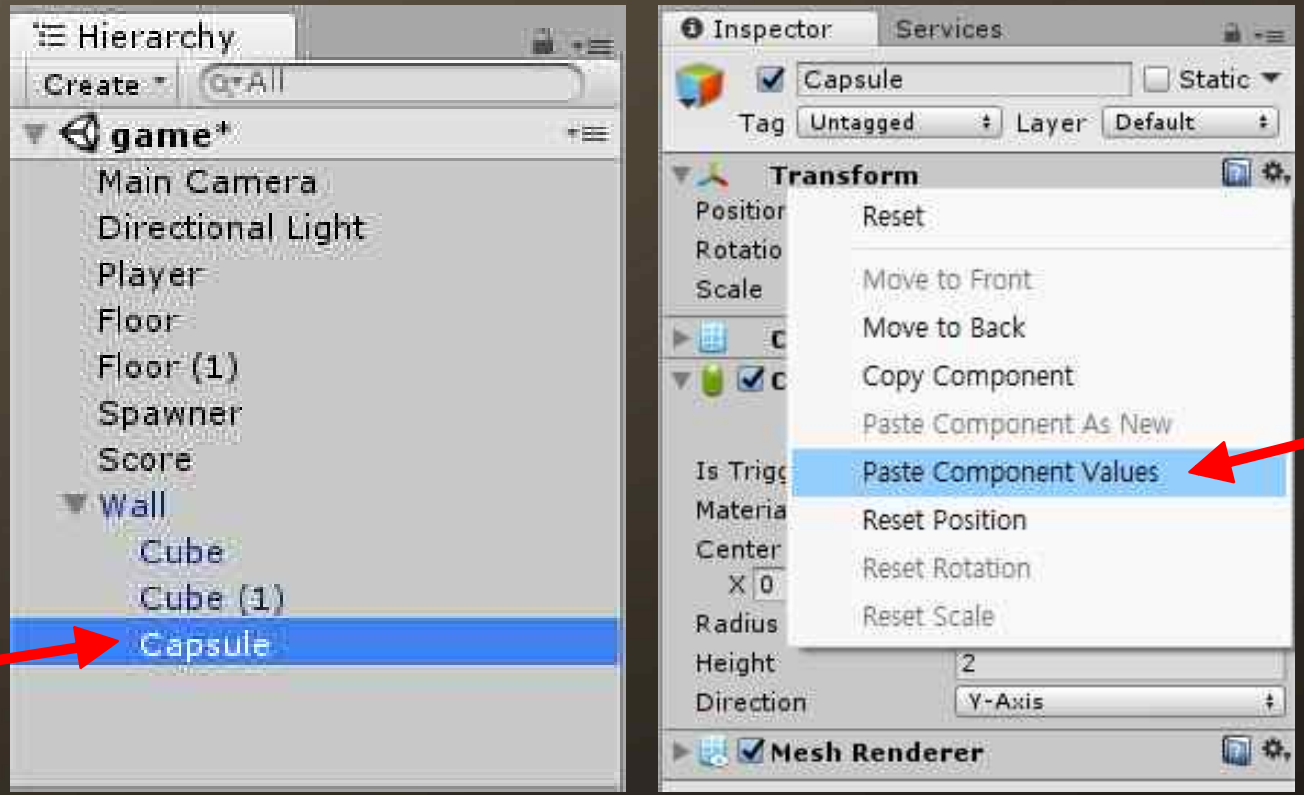
- GameObject > 3D Object > Capsule



시도6: 다양한 종류의 벽

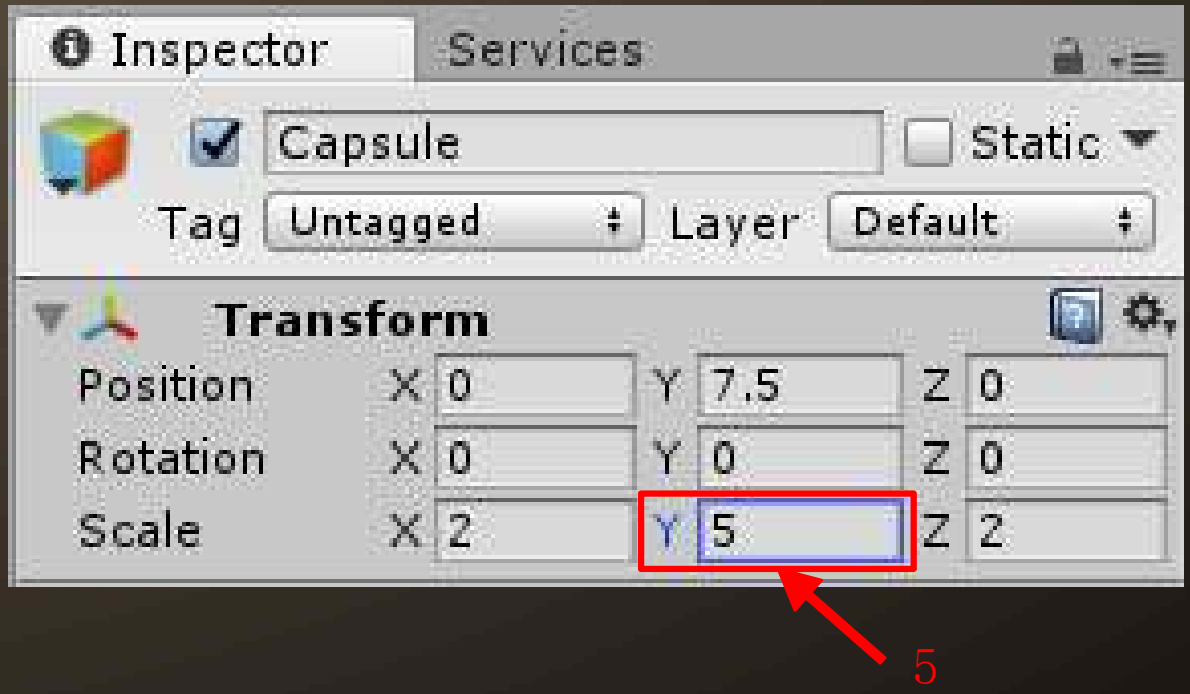

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule
1
Inspector
Services
Cube
Static
Tag
Untagged
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position
Rotation
Scale
Reset
Revert to Prefab
Move to Front
Move to Back
Copy Component
Paste Component As New
Paste Component Values
Reset Position
Reset Rotation
Reset Scale
2

시도6: 다양한 종류의 벽

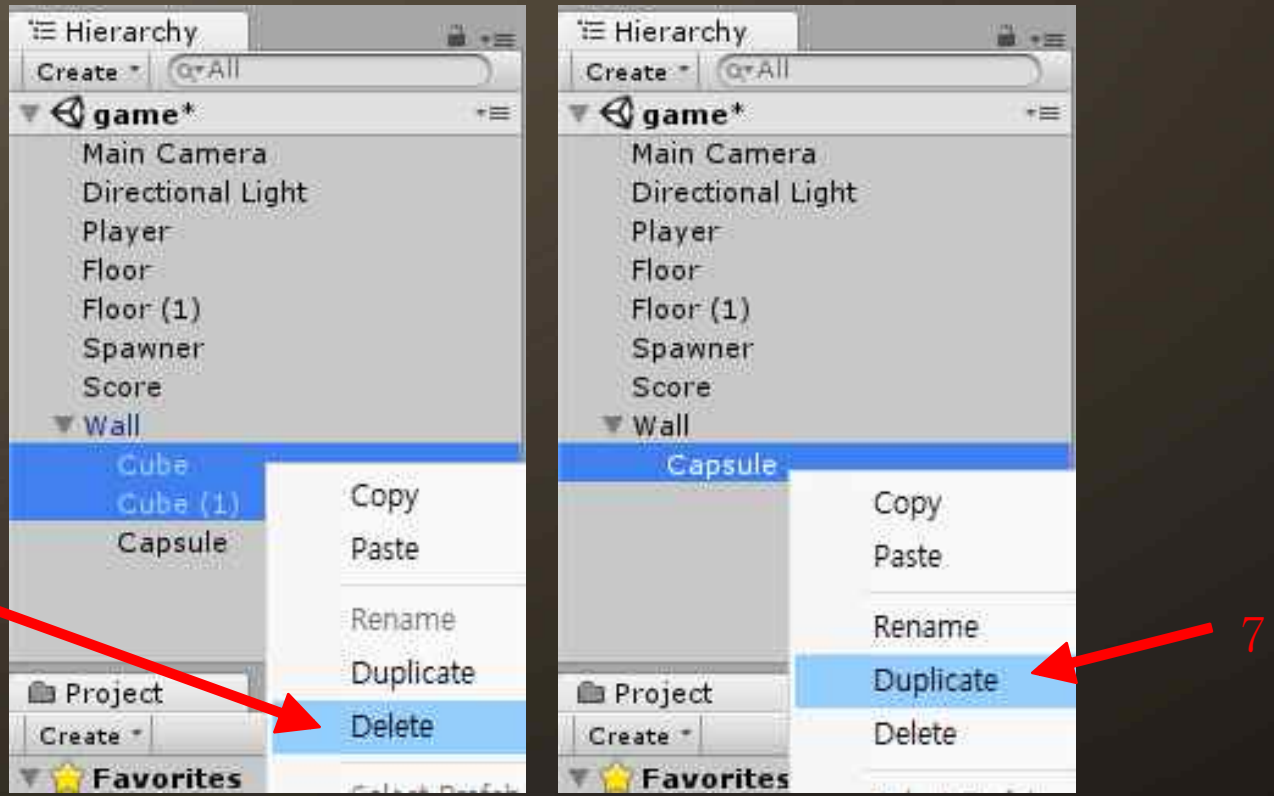

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule
3
Inspector
Services
Capsule
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Reset
Move to Front
Move to Back
Copy Component
Paste Component As New
Paste Component Values
Reset Position
Reset Rotation
Reset Scale
4
Is Trigg
Materia
Center
X 0
Radius
Height
2
Direction
Y-Axis
Mesh Renderer

시도6: 다양한 종류의 벽

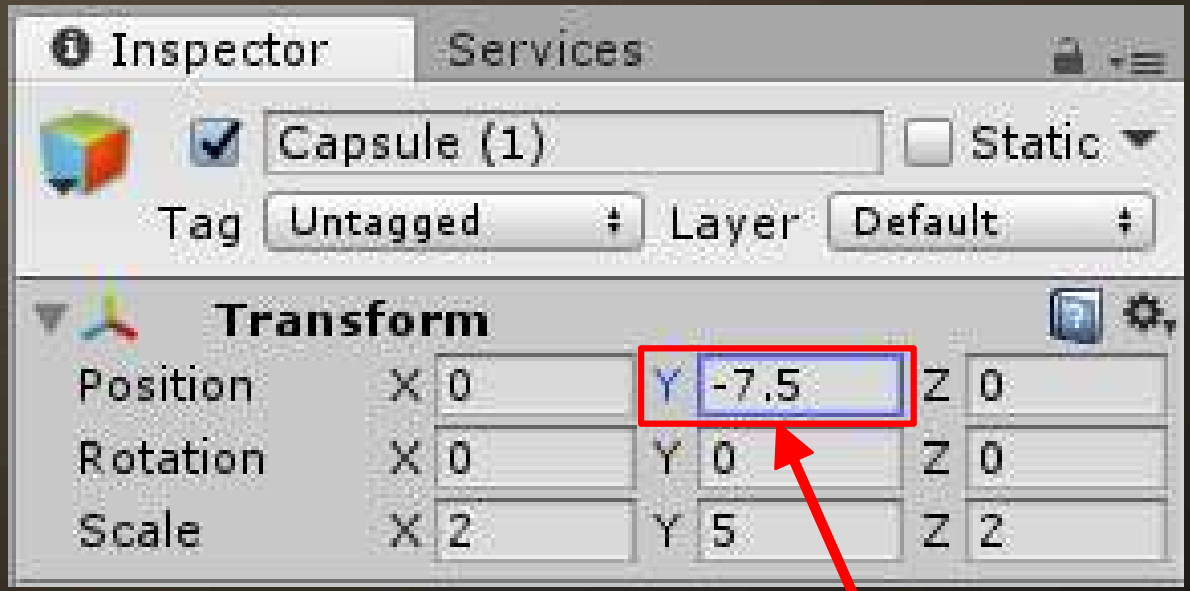

Hierarchy
Create
Q▾All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule


Inspector
Services
Capsule
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 7.5
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 2
Y 5
Z 2
5

시도6: 다양한 종류의 벽

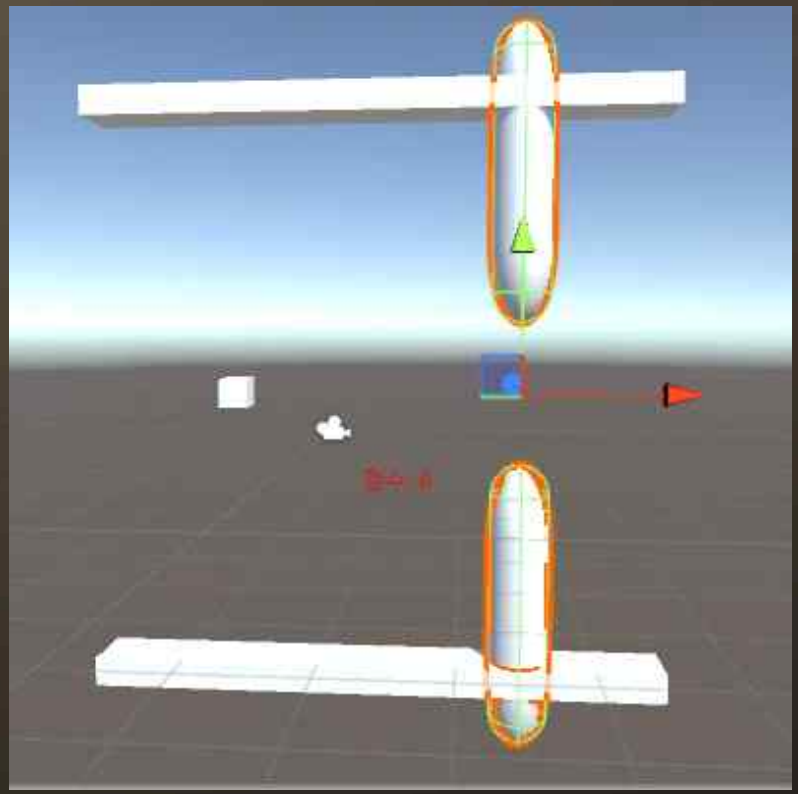

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule
Copy
Paste
Rename
Duplicate
Delete
Project
Create
Favorites
6
Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Capsule
Copy
Paste
Rename
Duplicate
Delete
Project
Create
Favorites
7

시도6: 다양한 종류의 벽

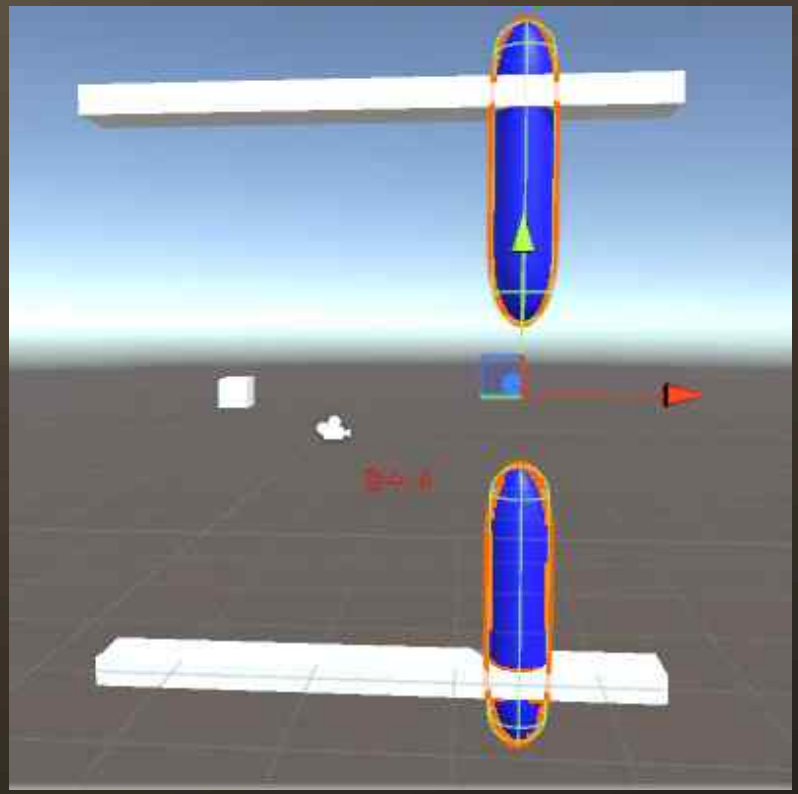


8

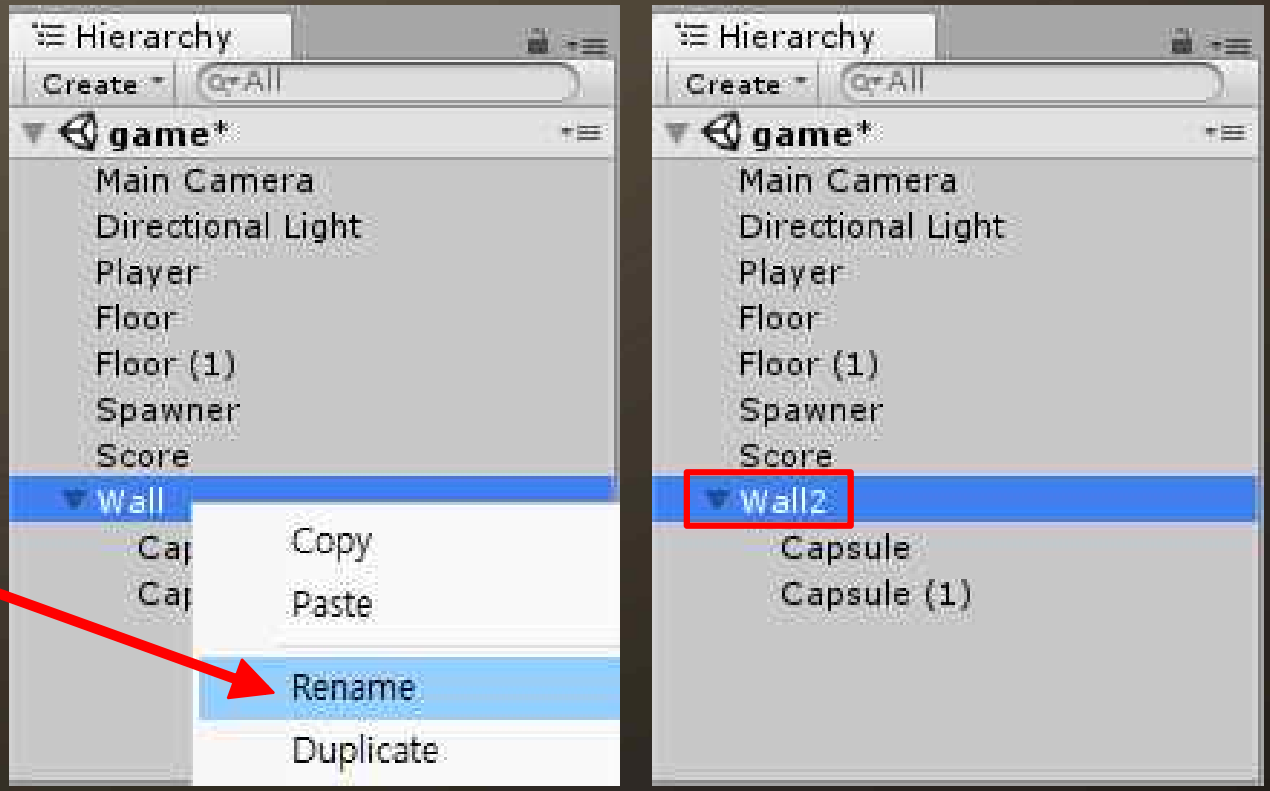
시도6: 다양한 종류의 벽



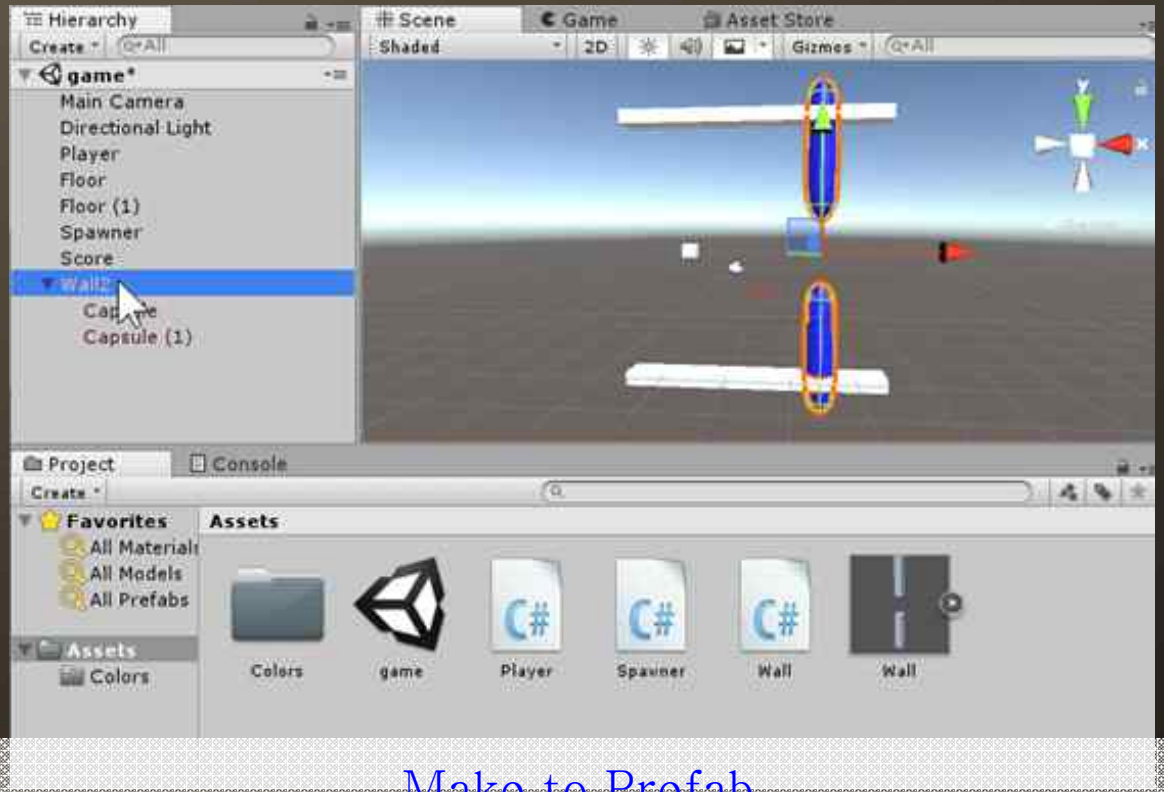
시도6: 다양한 종류의 벽



시도6: 다양한 종류의 벽

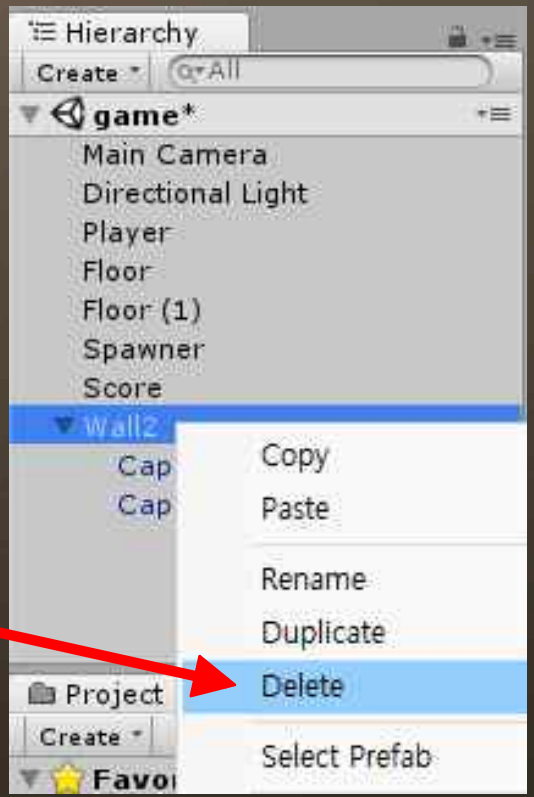

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cap
Cap
Copy
Paste
Rename
Duplicate
9
Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall2
Capsule
Capsule (1)

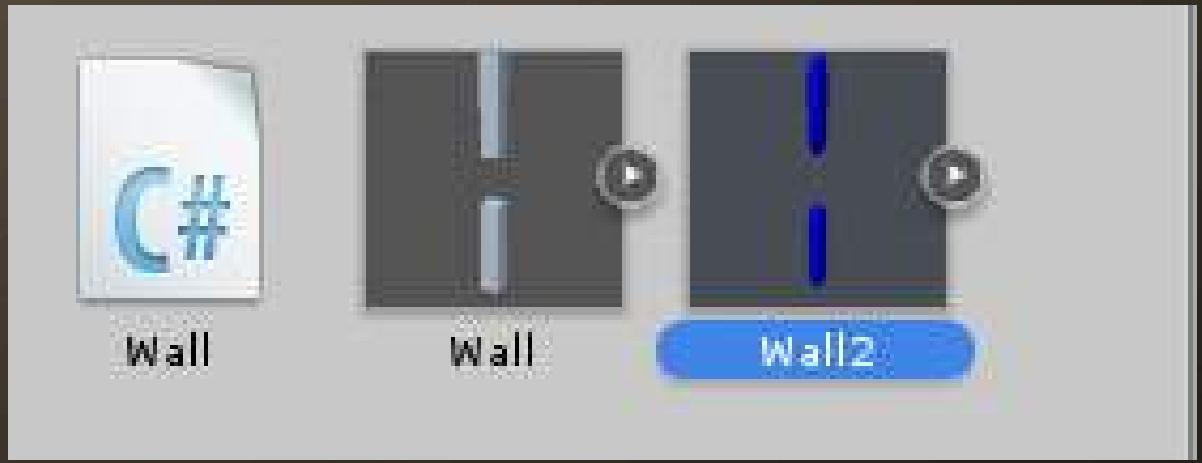
시도6: 다양한 종류의 벽



Make to Prefab

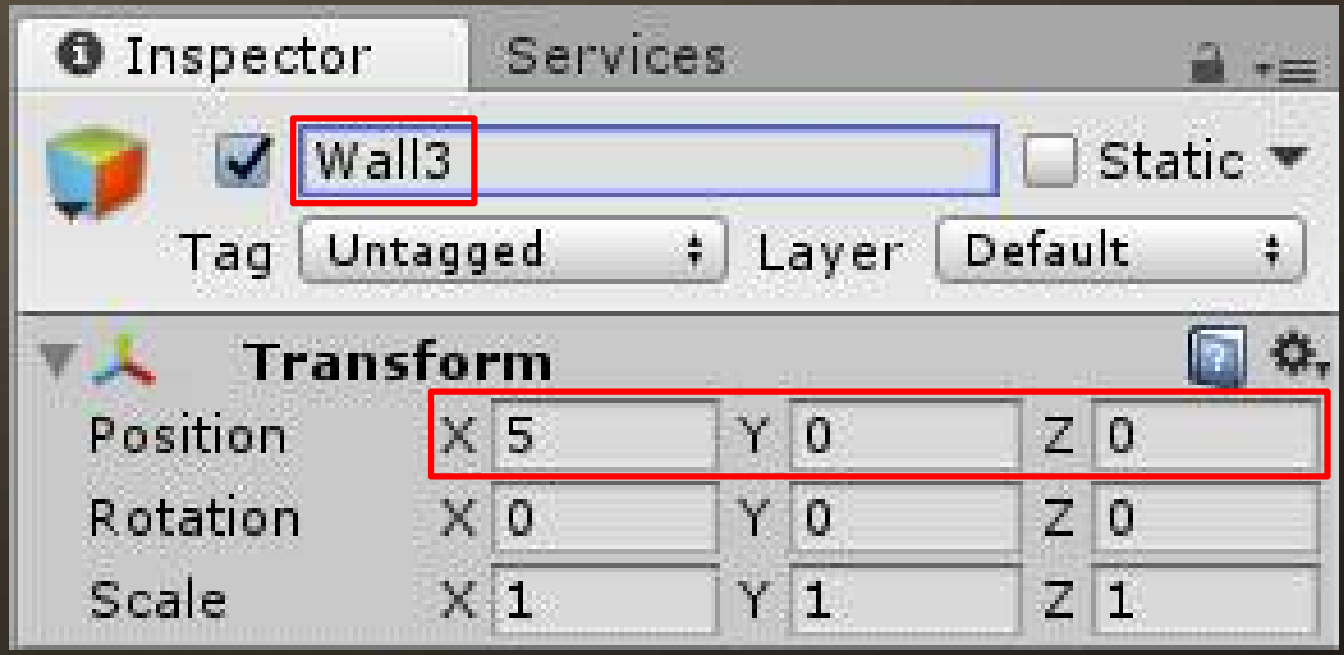
시도6: 다양한 종류의 벽


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall2
Cap
Cap
Copy
Paste
Rename
Duplicate
Delete
Select Prefab
Project
Create
Favo


Wall
Wall
Wall2

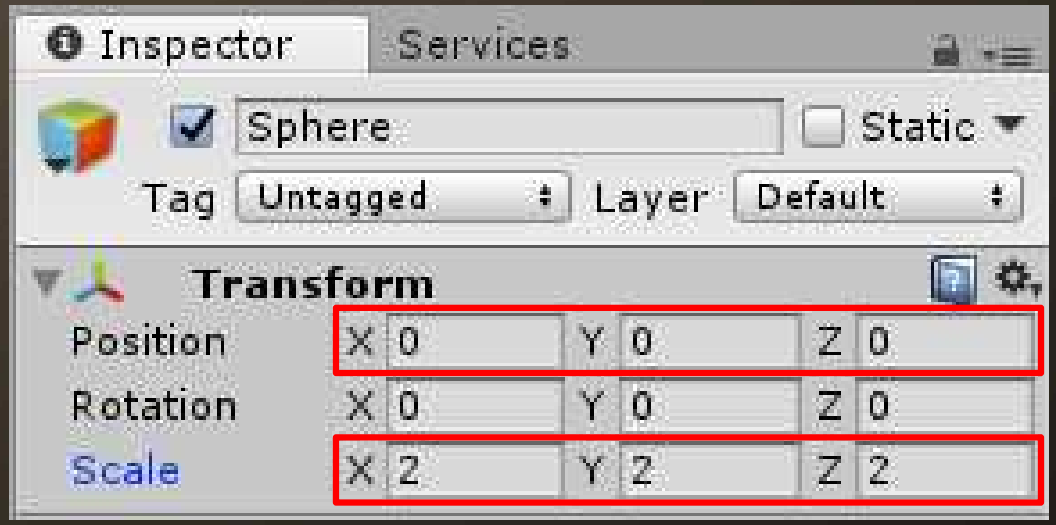
시도6: 다양한 종류의 벽

. GameObject > Create Empty



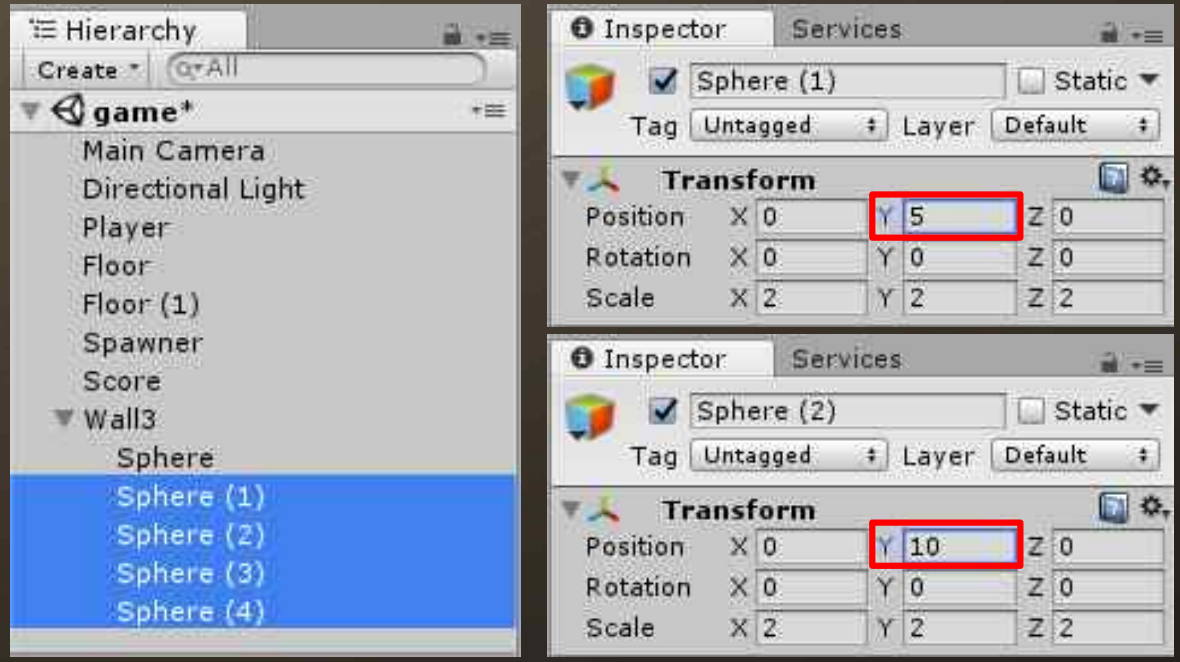
시도6: 다양한 종류의 벽

- GameObject > 3D Object > Sphere



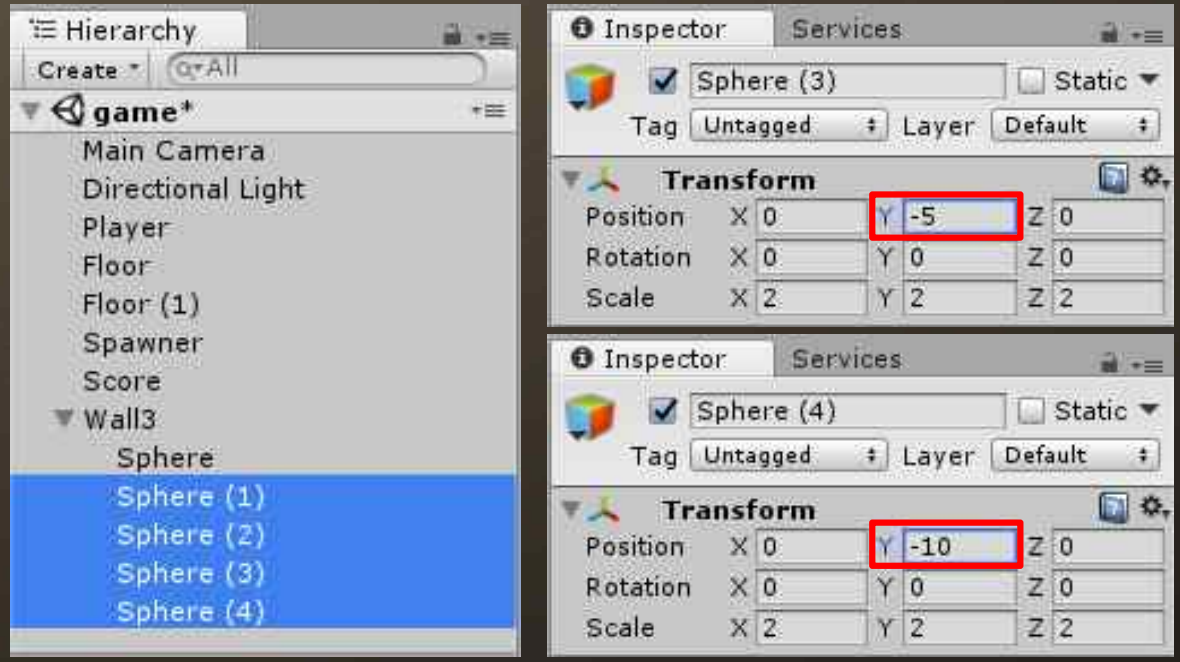
시도6: 다양한 종류의 벽

. Sphere > Duplicate

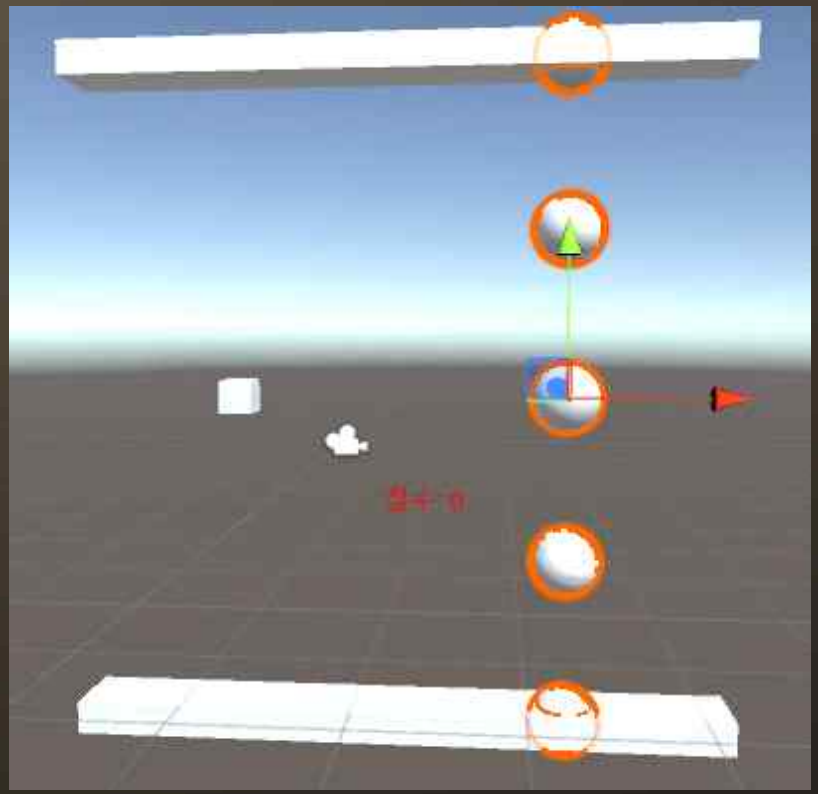


시도6: 다양한 종류의 벽

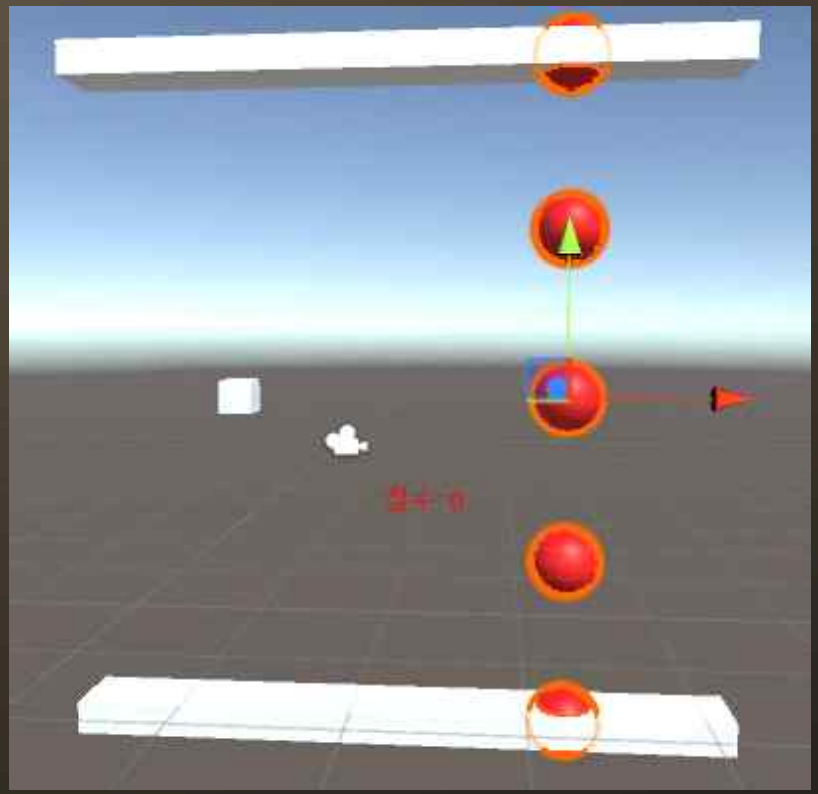
. Sphere > Duplicate



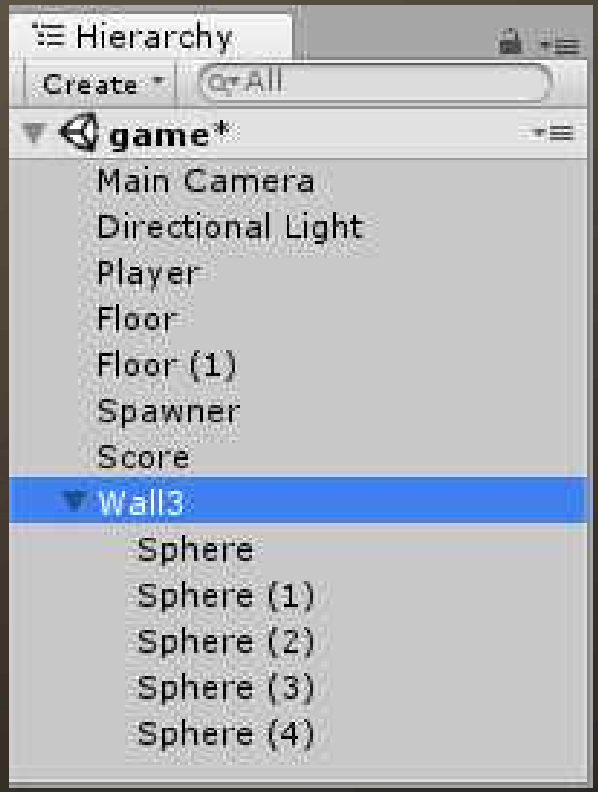
시도6: 다양한 종류의 벽

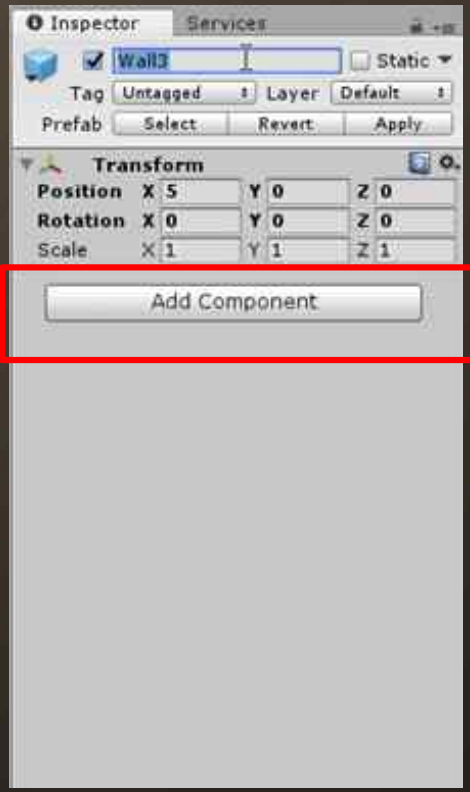


시도6: 다양한 종류의 벽

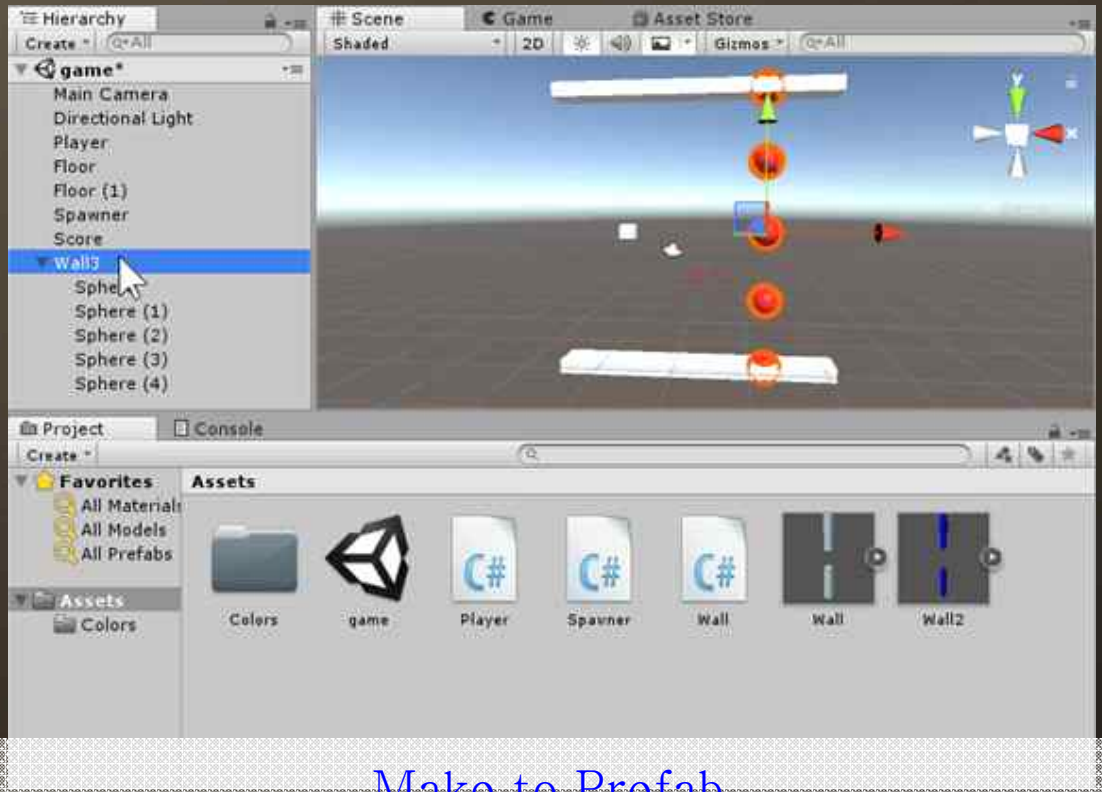


시도6: 다양한 종류의 벽


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall3
Sphere
Sphere (1)
Sphere (2)
Sphere (3)
Sphere (4)

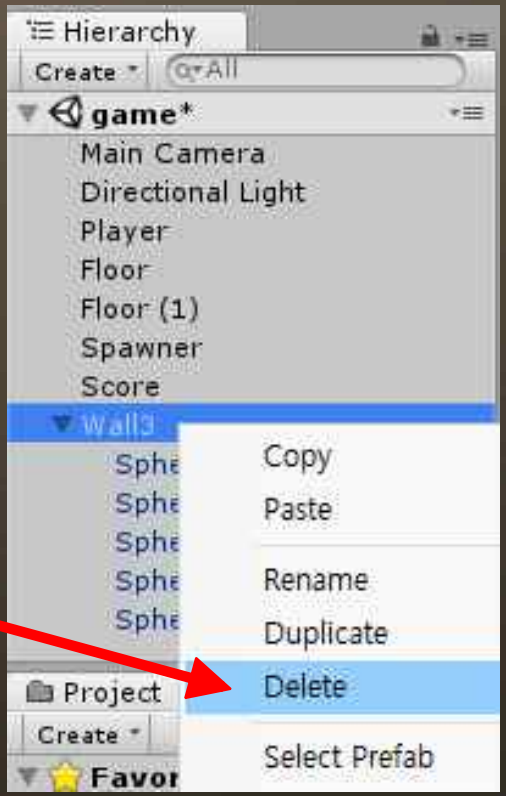

Inspector
Services
Wall3
Static
Tag
Untagged
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position X 5 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Add Component

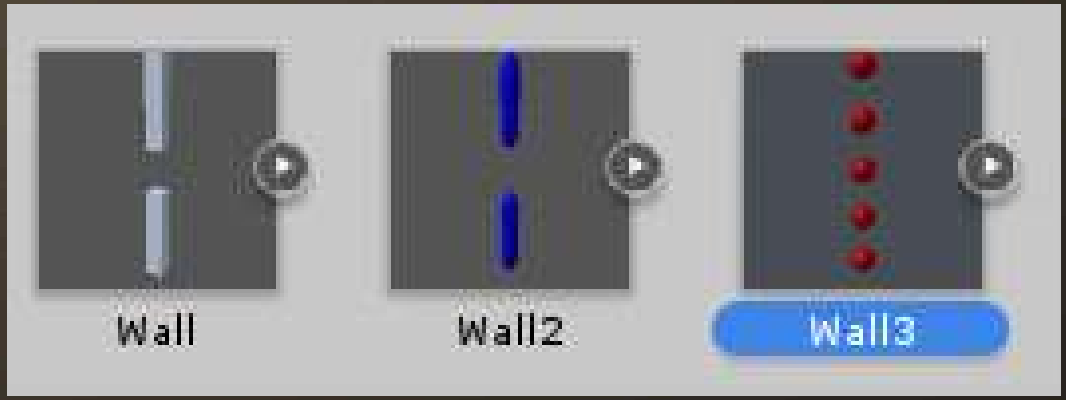
시도6: 다양한 종류의 벽



Make to Prefab

시도6: 다양한 종류의 벽


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall3
Copy
Paste
Rename
Duplicate
Delete
Select Prefab
Project
Create


Wall
Wall2
Wall3

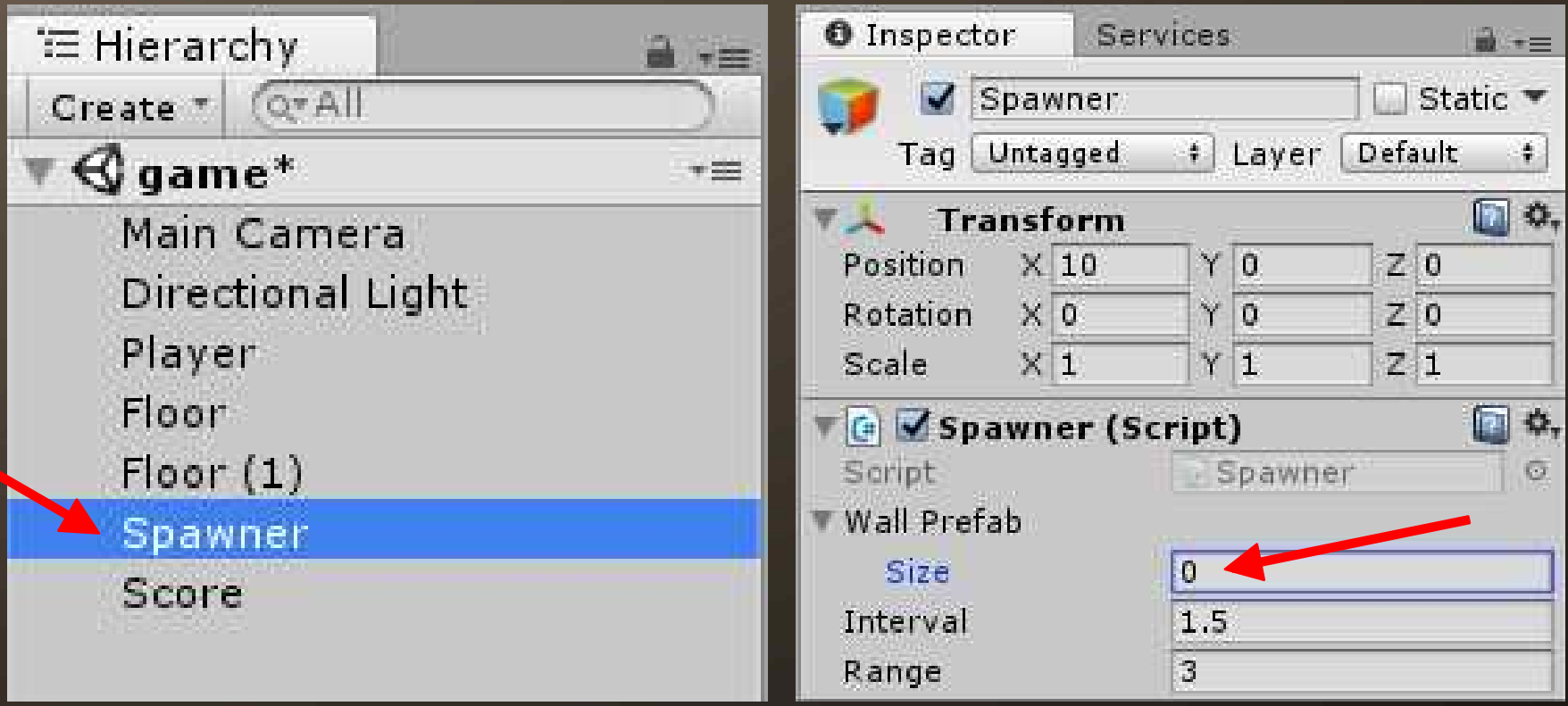
시도6: 다양한 종류의 벽

```
public class Spawner : MonoBehaviour {

    public GameObject[] wallPrefab;
    public float interval = 1.5f;
    public float range = 3;
    float term;
```

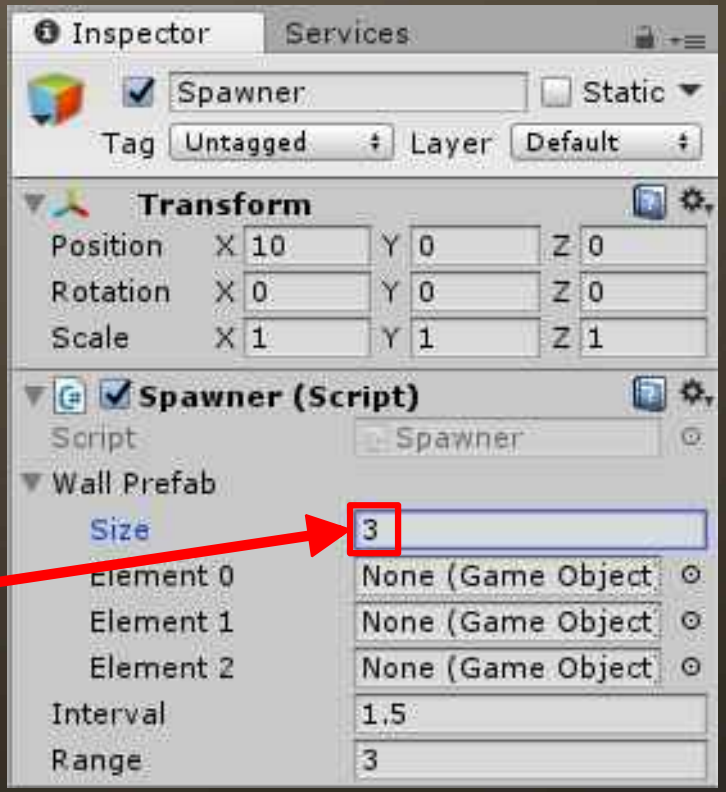
시도6: 다양한 종류의 벽

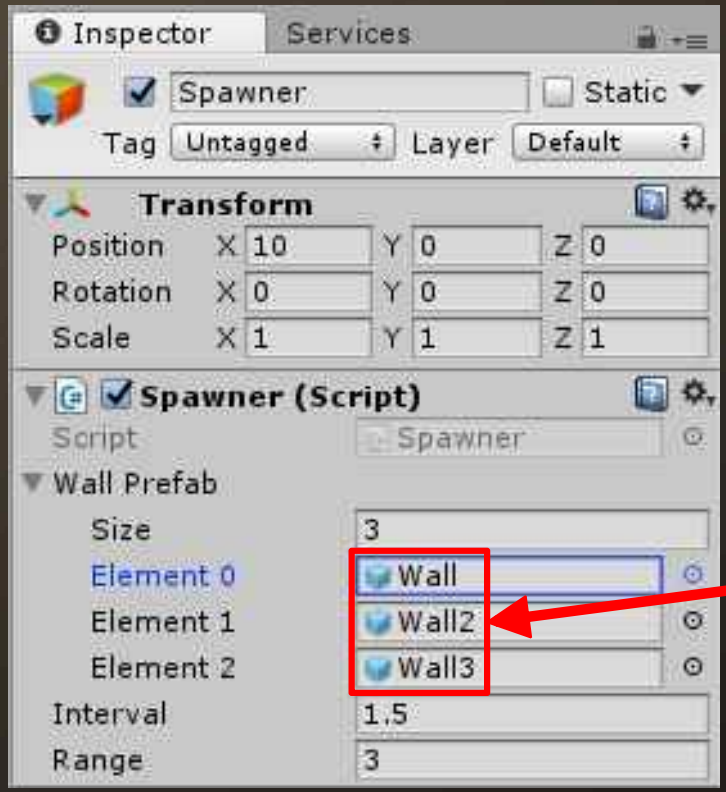
```
        // Update is called once per frame
        void Update () {
                term += Time.deltaTime;
                if (term >= interval)
                {
                        Vector3 pos = transform.position;
                        pos.y += Random.Range(-range, range);
                        int wallType = Random.Range(0, wallPrefab.Length);
                        Instantiate(wallPrefab[wallType], pos,
transform.rotation);
                        term -= interval;
                }
        }
}
```

시도6: 다양한 종류의 벽

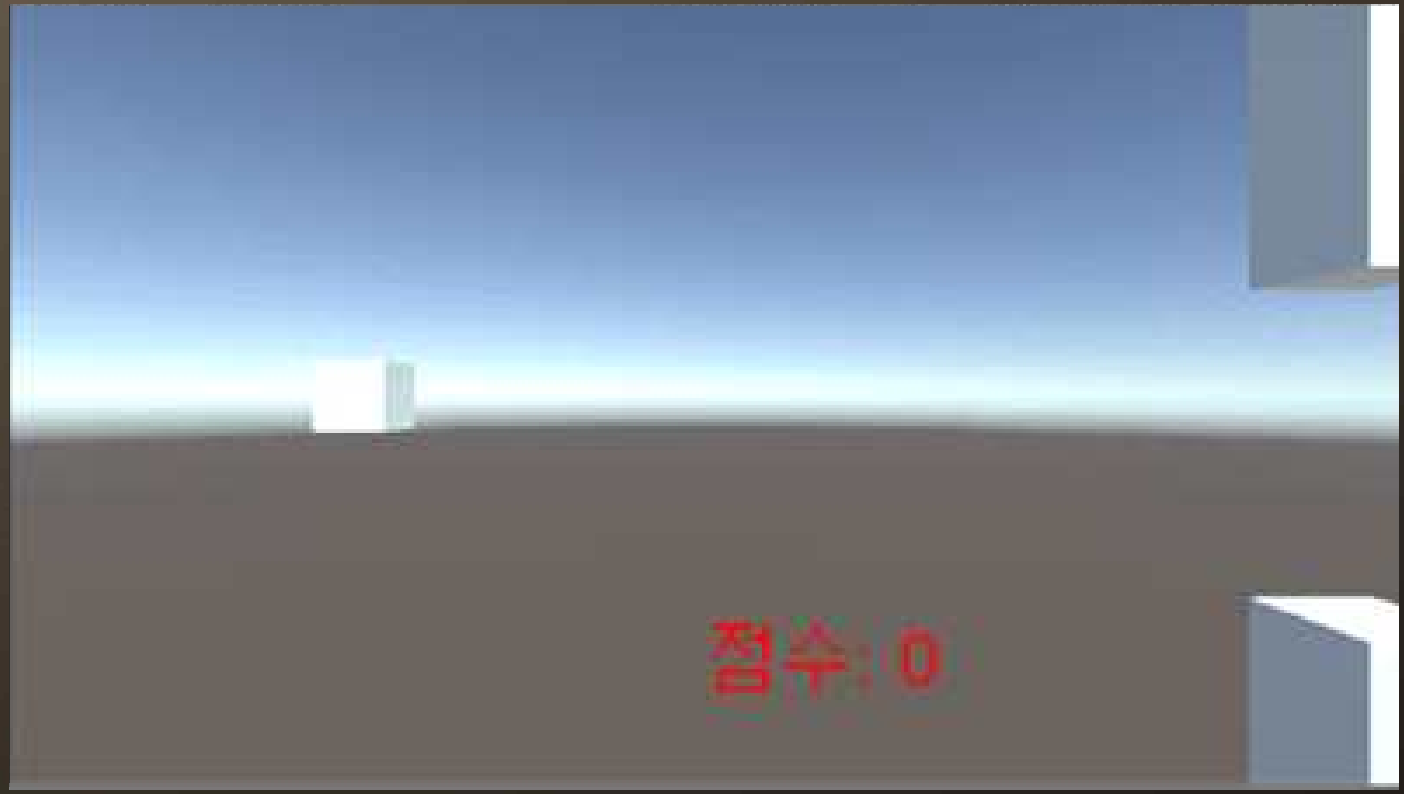

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Position
X 10
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Spawner (Script)
Script
Spawner
Wall Prefab
Size
0
Interval
1.5
Range
3

시도6: 다양한 종류의 벽


Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Position
X 10
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Spawner (Script)
Script
Spawner
Wall Prefab
Size
3
Element 0
None (Game Object
Element 1
None (Game Object
Element 2
None (Game Object
Interval
1.5
Range
3


Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Position
X 10
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Spawner (Script)
Script
Spawner
Wall Prefab
Size
3
Element 0
Wall
Element 1
Wall2
Element 2
Wall3
Interval
1.5
Range
3

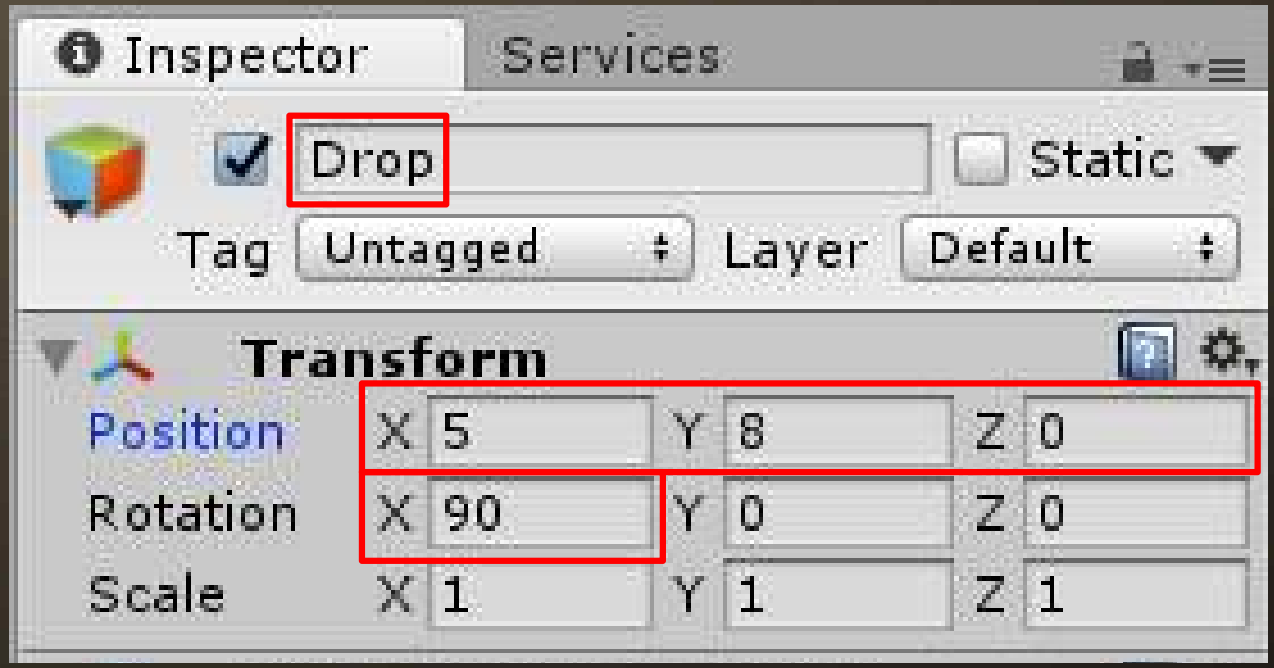
시도6: 다양한 종류의 벽



Try 7
위에서 떨어지는 장애물

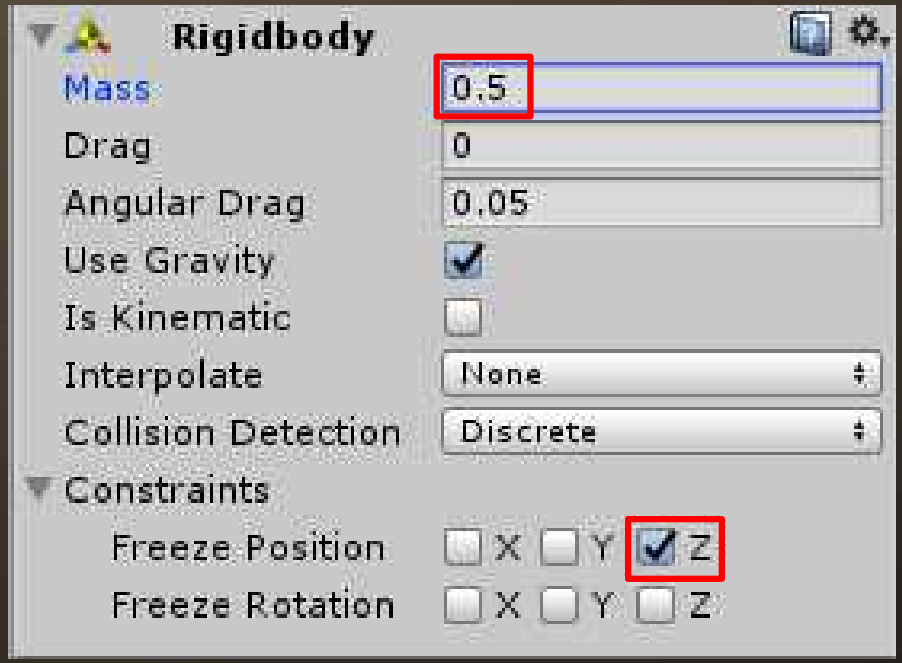
시도7: 위에서 떨어지는 장애물

. GameObject > 3D Object > Cylinder

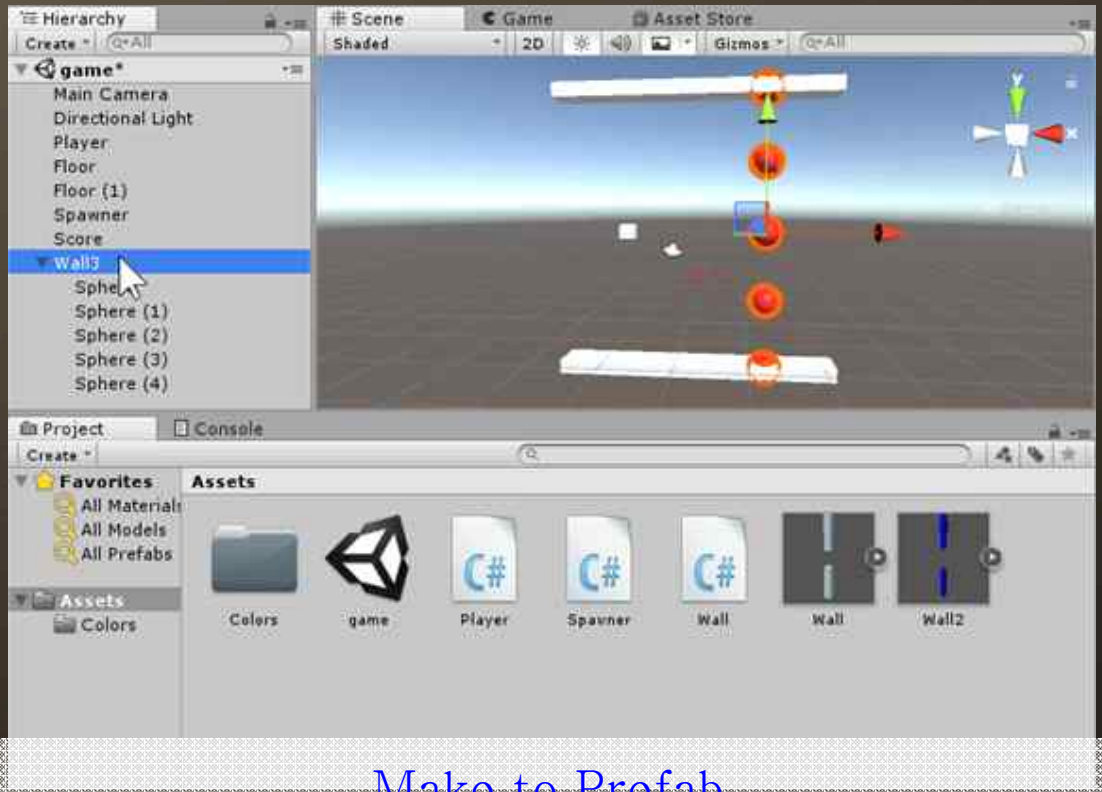


시도7: 위에서 떨어지는 장애물

. Component > Physics > Rigidbody

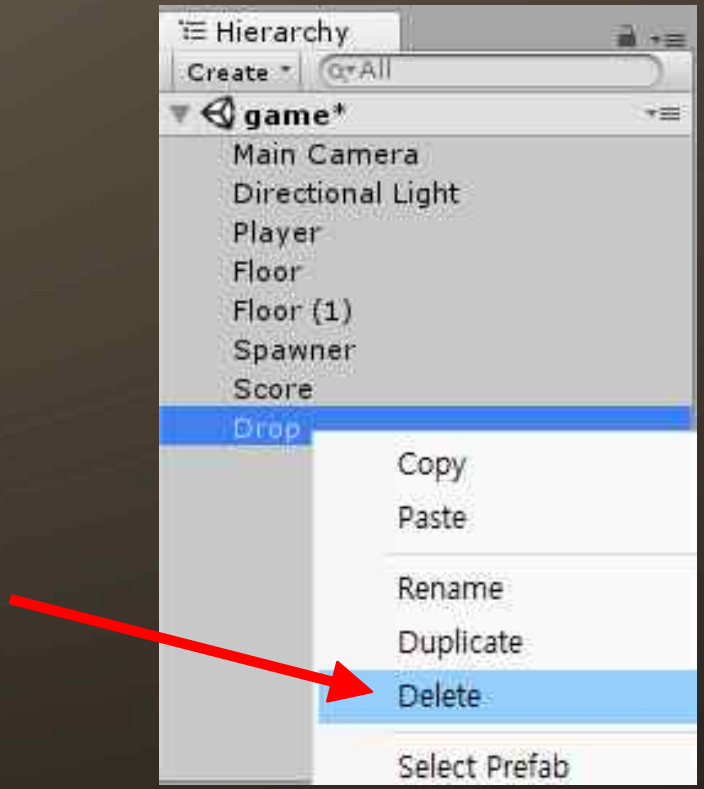


시도7: 위에서 떨어지는 장애물



Make to Prefab

시도7: 위에서 떨어지는 장애물



시도7: 위에서 떨어지는 장애물

Spawner.cs | MiniGame | Spawner | Start()

```
public class Spawner : MonoBehaviour {

    public GameObject[] wallPrefab;
    public GameObject   dropPrefab;
    public float interval = 1.5f;
    public float range = 3;
    float term;
```

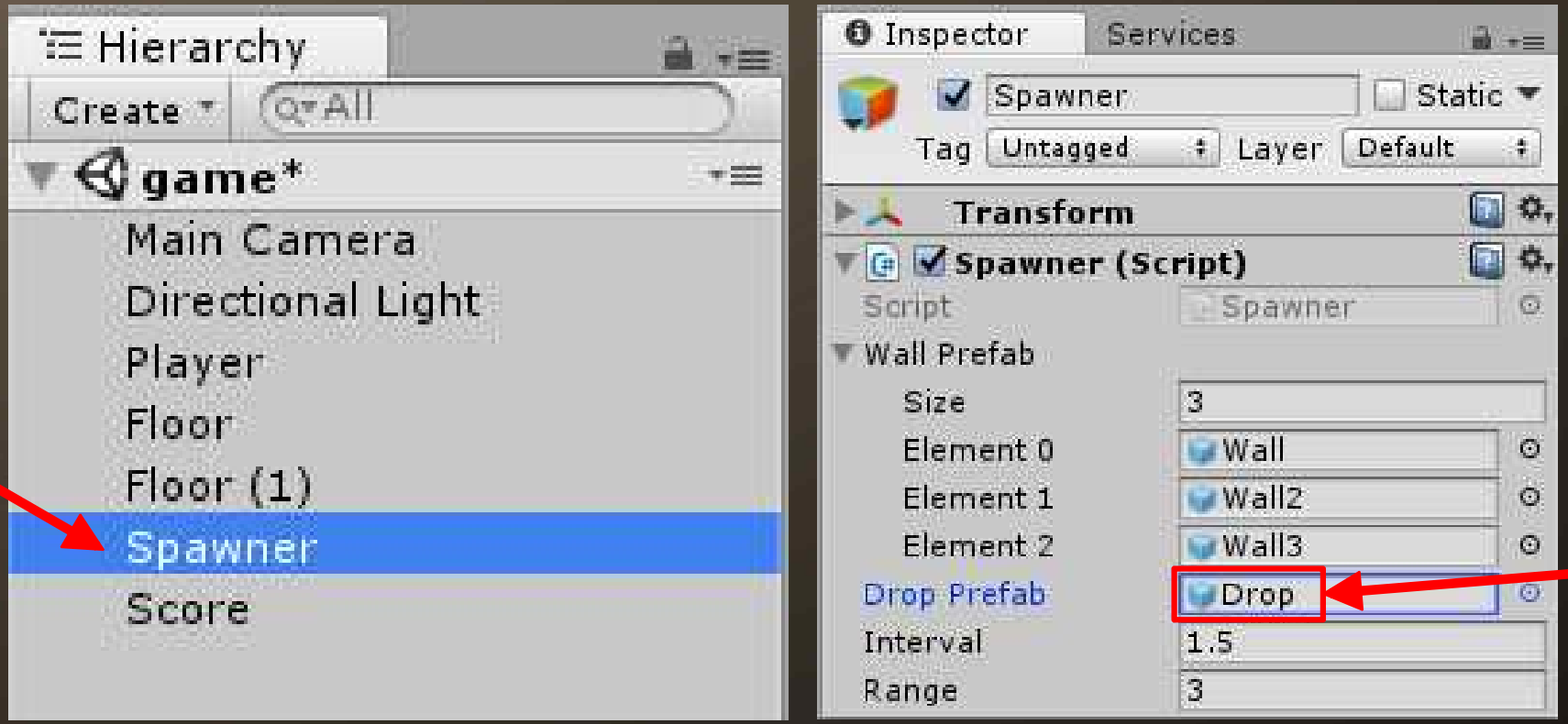
시도7: 위에서 떨어지는 장애물

Spawner.cs

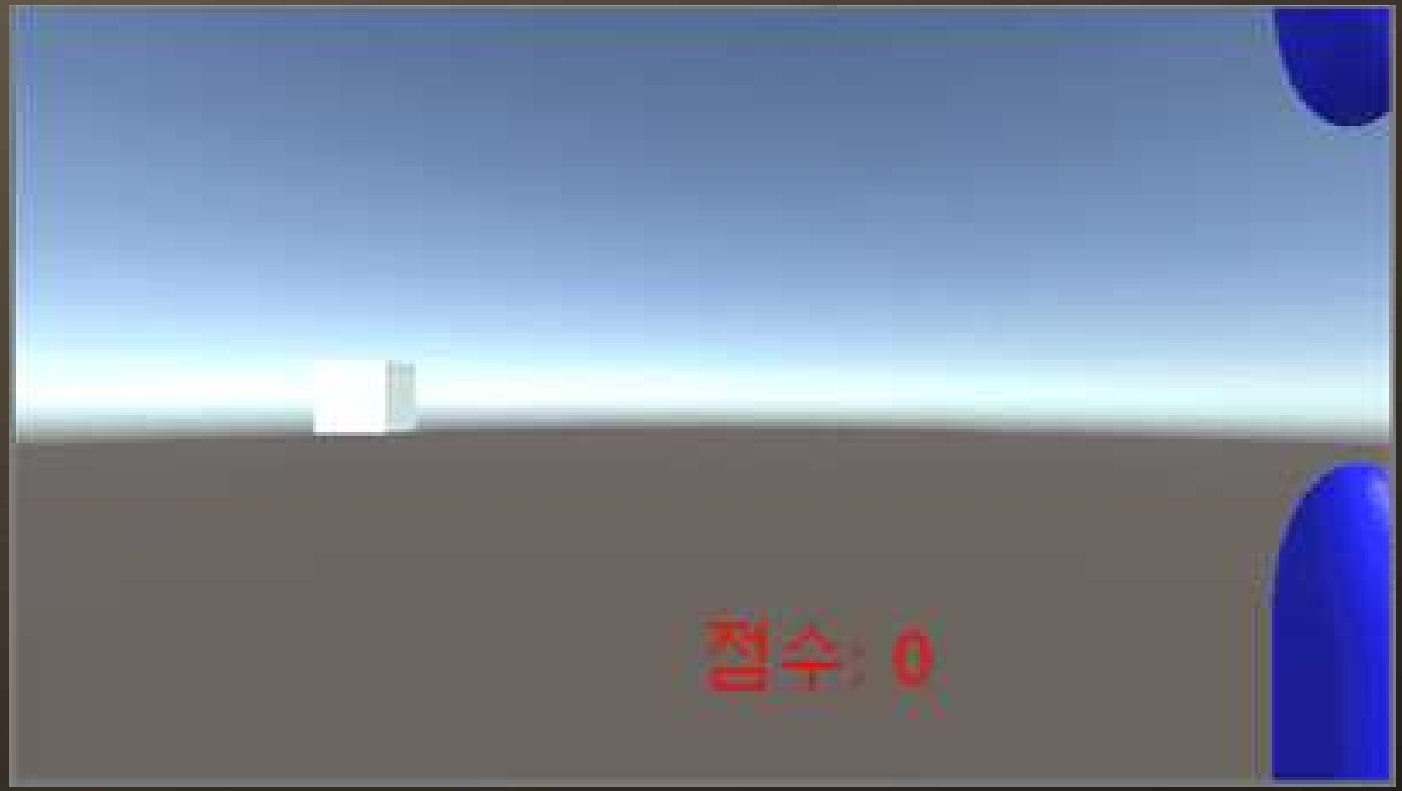
MiniGame | Spawner | Update()

```
        // Update is called once per frame
        void Update () {
                term += Time.deltaTime;
                if (term >= interval)
                {
                        Vector3 pos = transform.position;
                        pos.y += Random.Range(-range, range);
                        int wallType = Random.Range(0, wallPrefab.Length);
                        Instantiate(wallPrefab[wallType], pos,
transform.rotation);
                        if (Random.Range(0, 2) == 0) // 50%의 확률로
                                Instantiate(dropPrefab); // 떨어지는 장애물 생성
                        term -= interval;
                }
```

시도7: 위에서 떨어지는 장애물



시도7: 위에서 떨어지는 장애물



감사합니다.

김태완
010-7614-4119
melvin@kbu.ac.kr