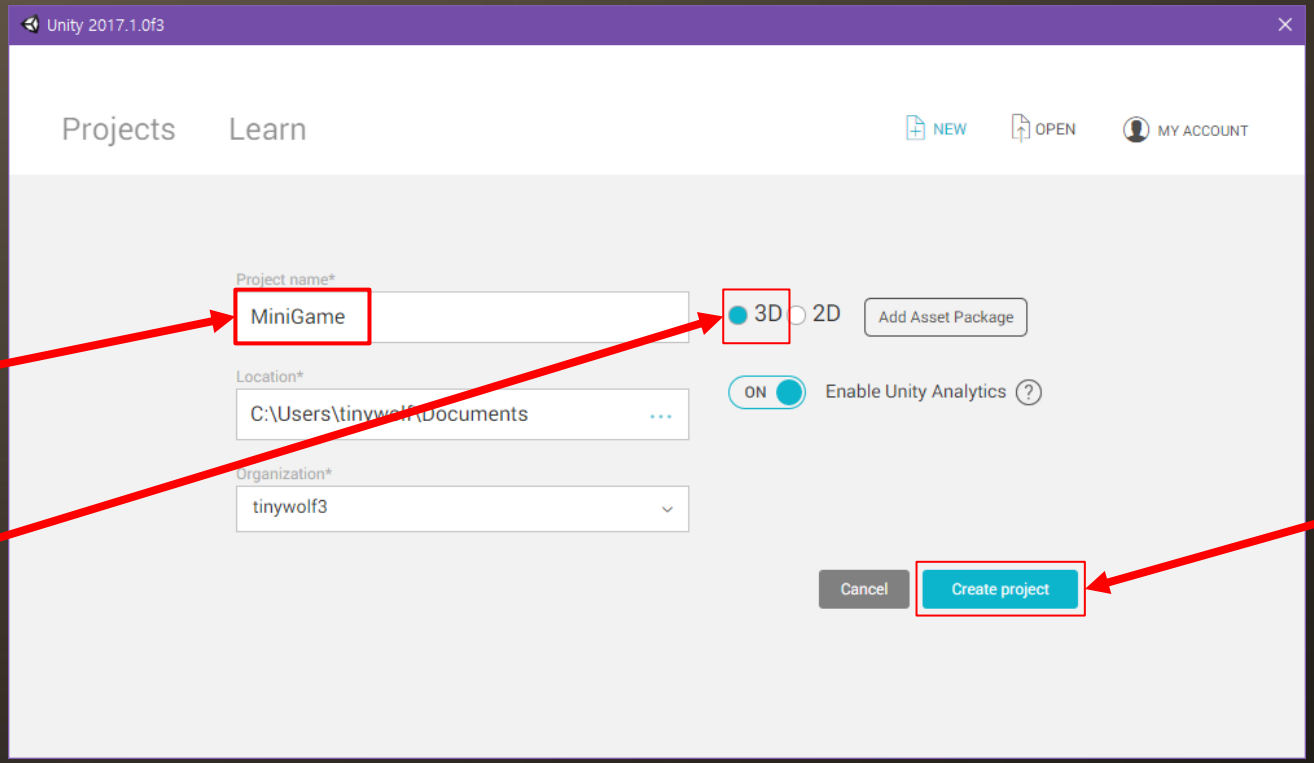
미니 게임 제작

DELICIOUS GAMES

Mini Game Project

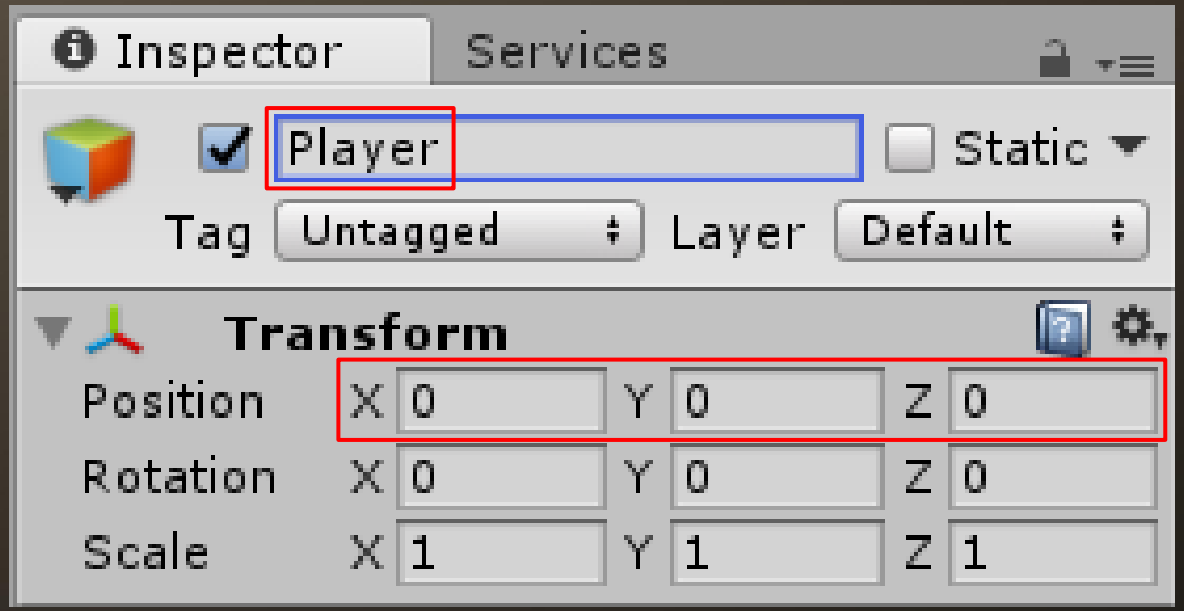
DELICIOUS GAMES

프로젝트 생성



오브젝트 추가

1. GameObject > 3D Object > Cube



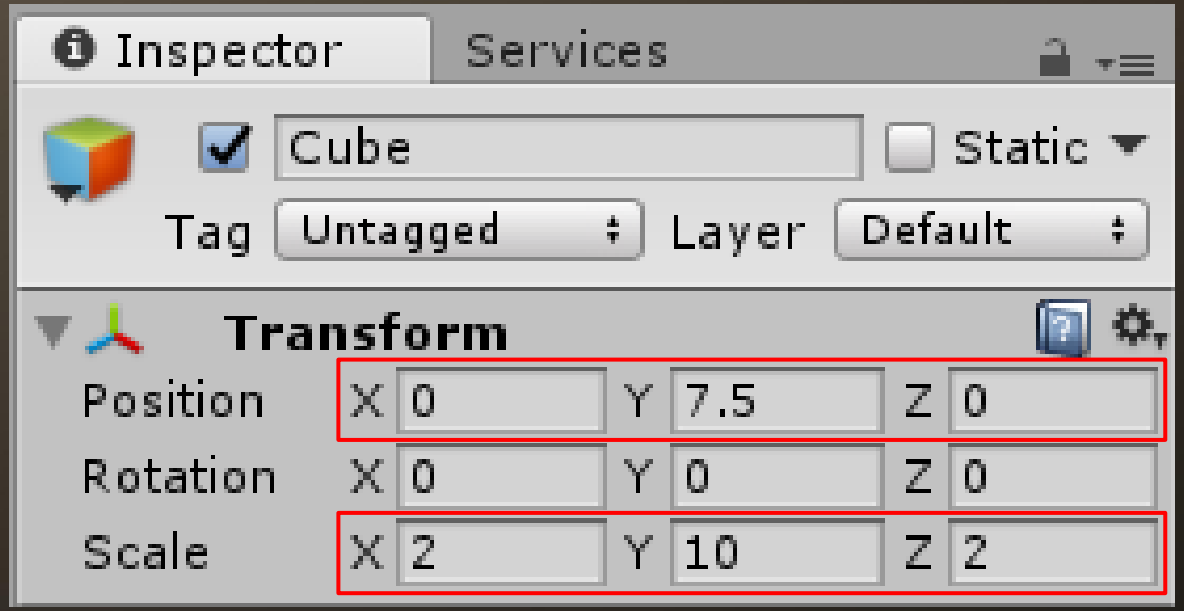
장애물

DELICIOUS GAMES

Obstacles

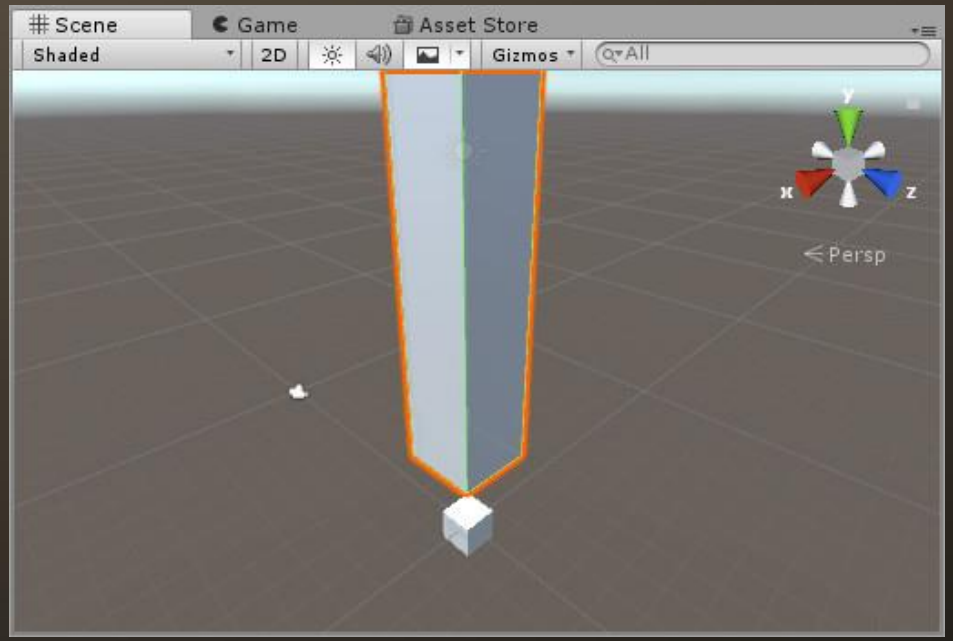
장애물 벽 추가

3. GameObject > 3D Object > Cube



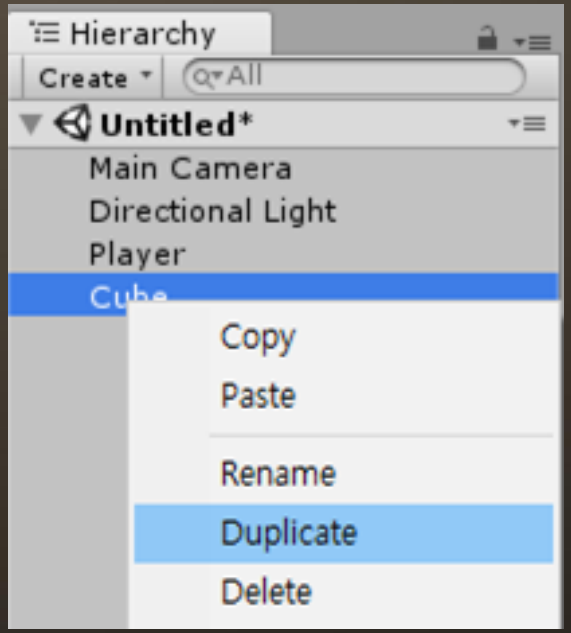
장애물 벽 추가

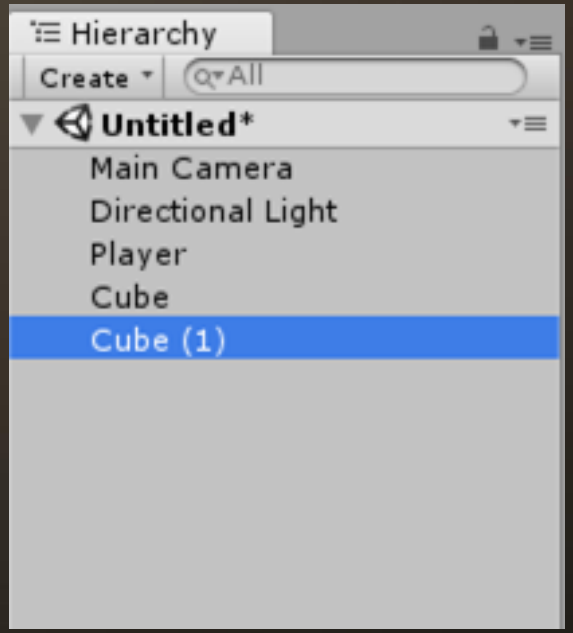
3. GameObject > 3D Object > Cube



벽 복제

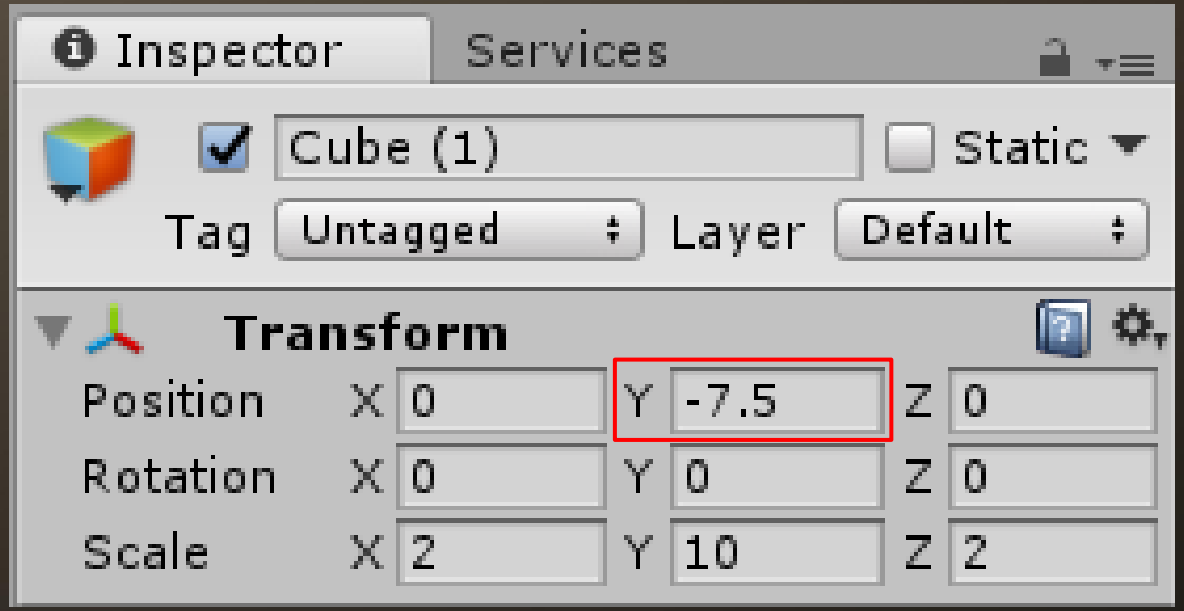
4. Cube > Duplicate





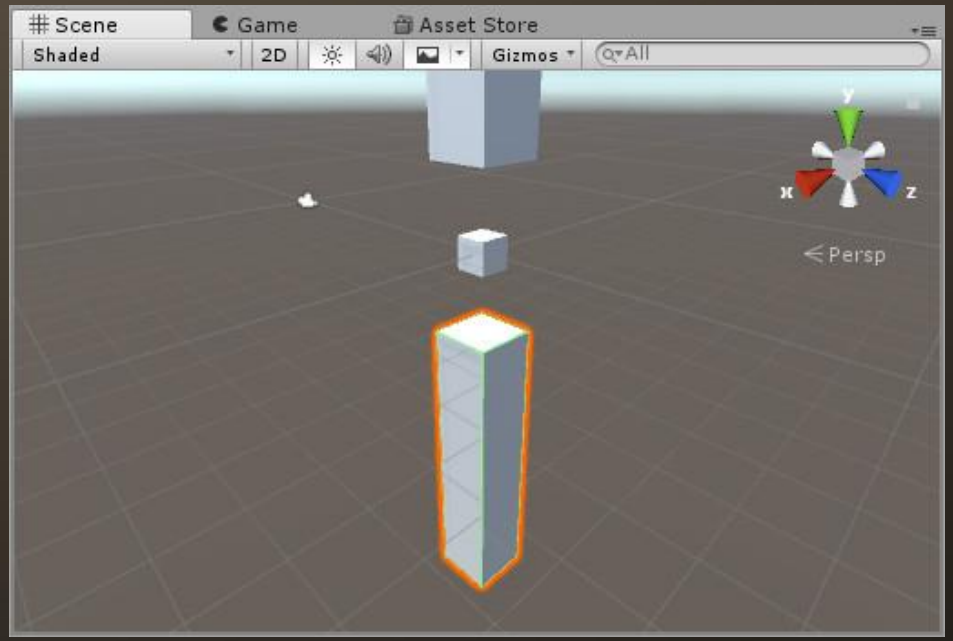
벽 배치

4. Cube > Duplicate



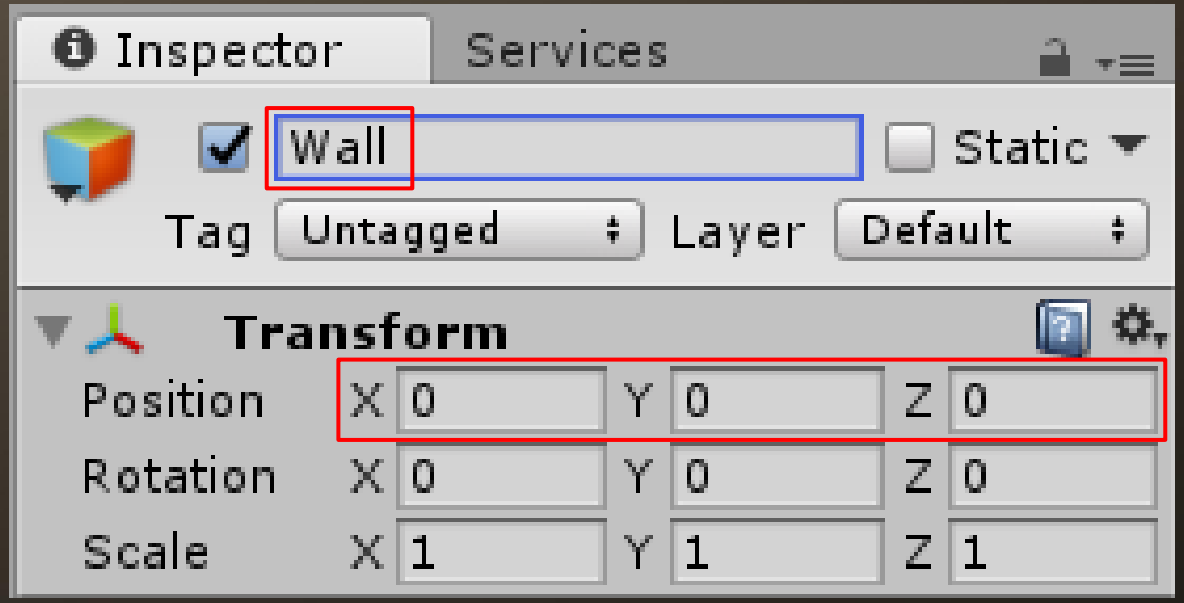
벽 배치

4. Cube > Duplicate



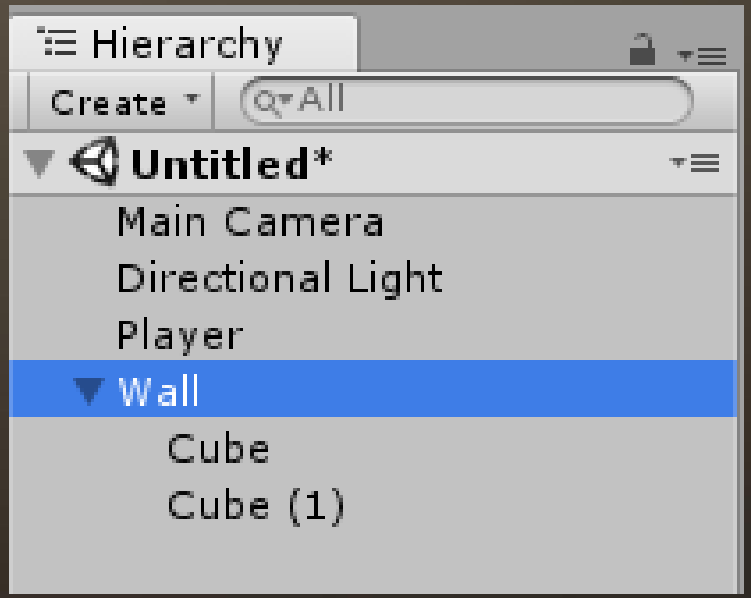
빈 오브젝트 추가

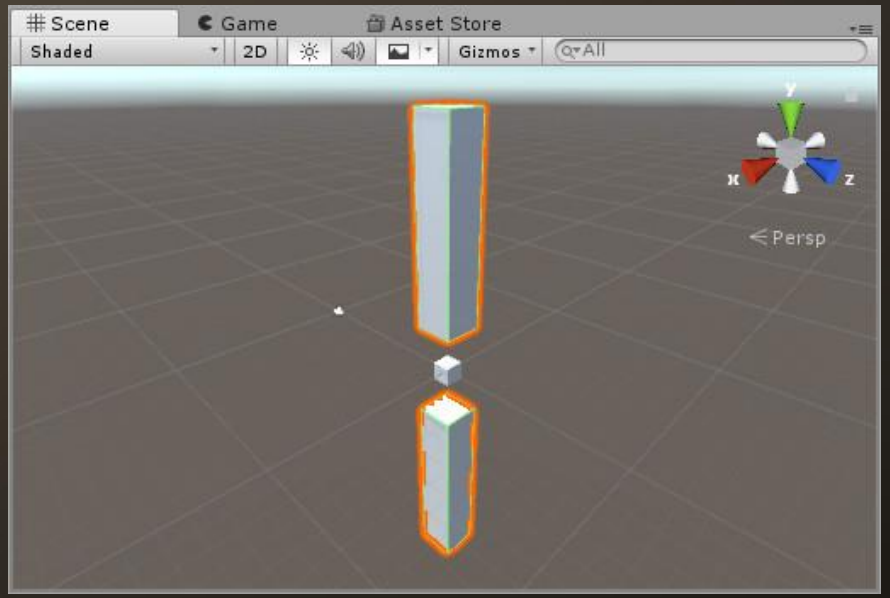
5. GameObject > Create Empty



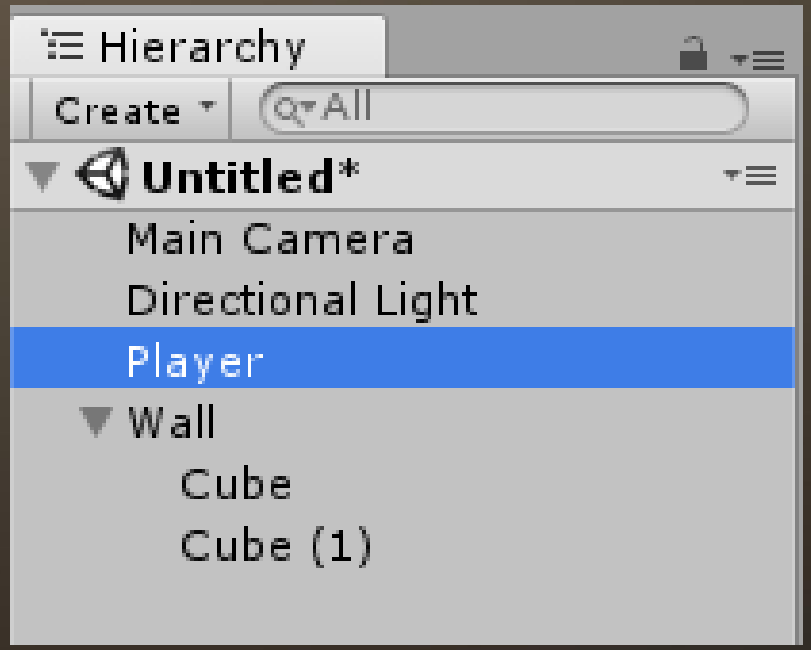
벽 오브젝트 조정

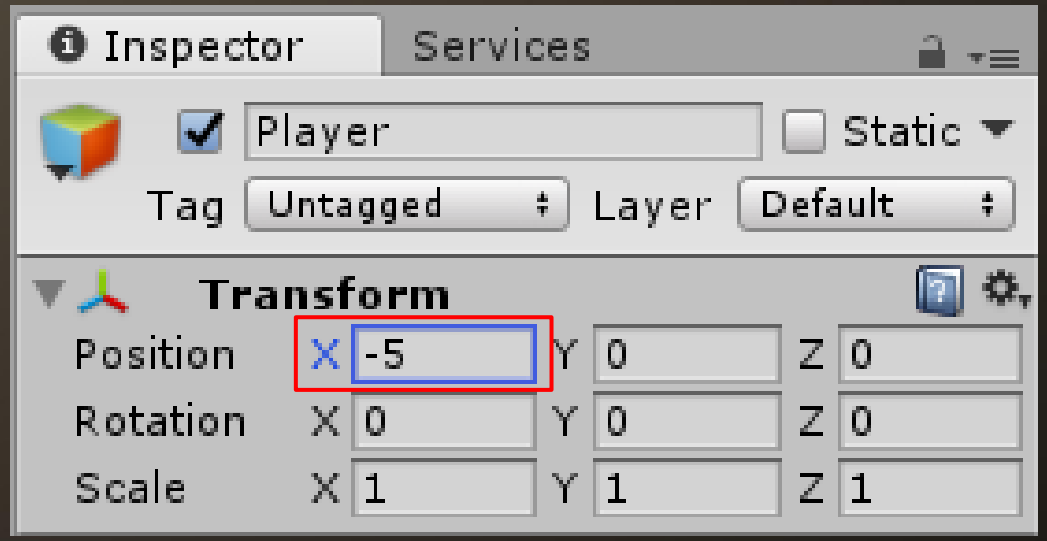
5. GameObject > Create Empty



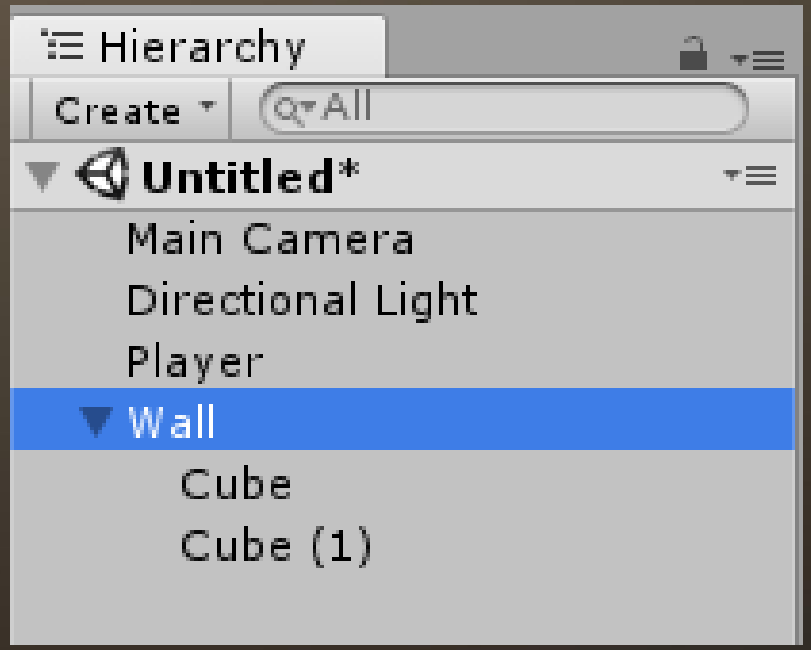


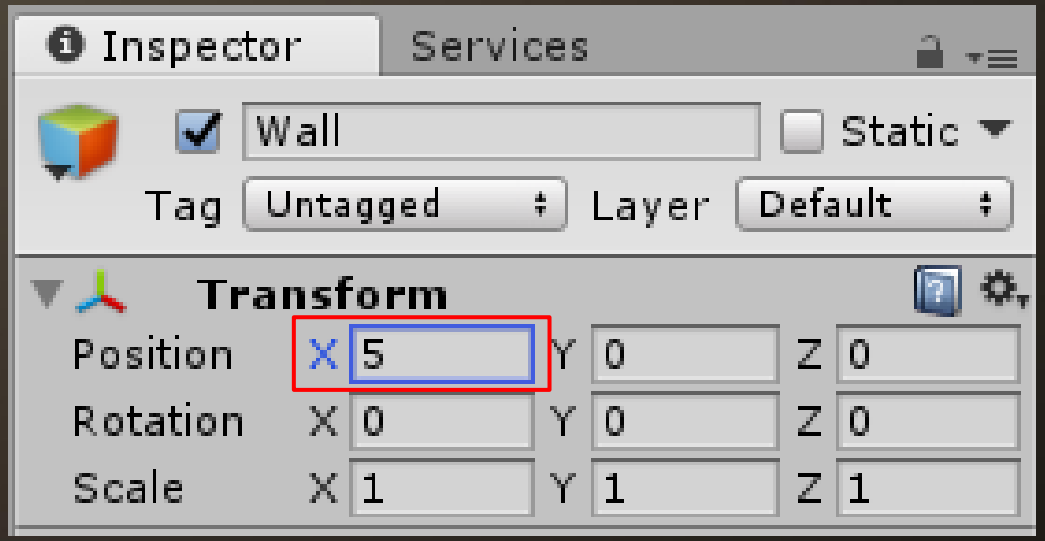
플레이어와 벽의 위치 조정



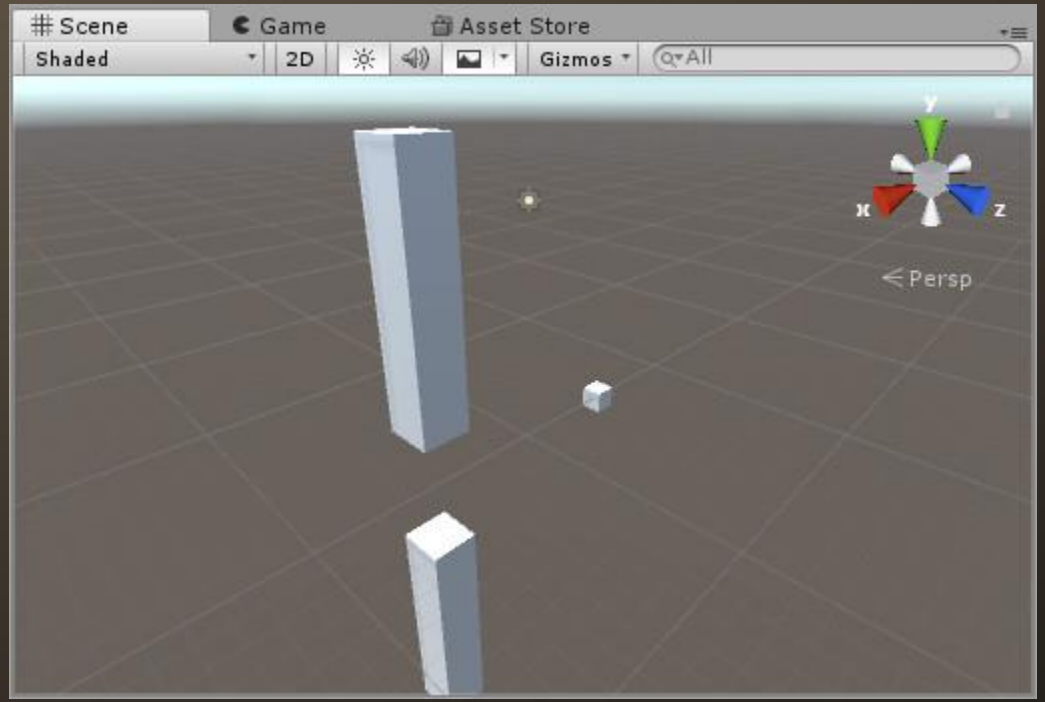


플레이어와 벽의 위치 조정

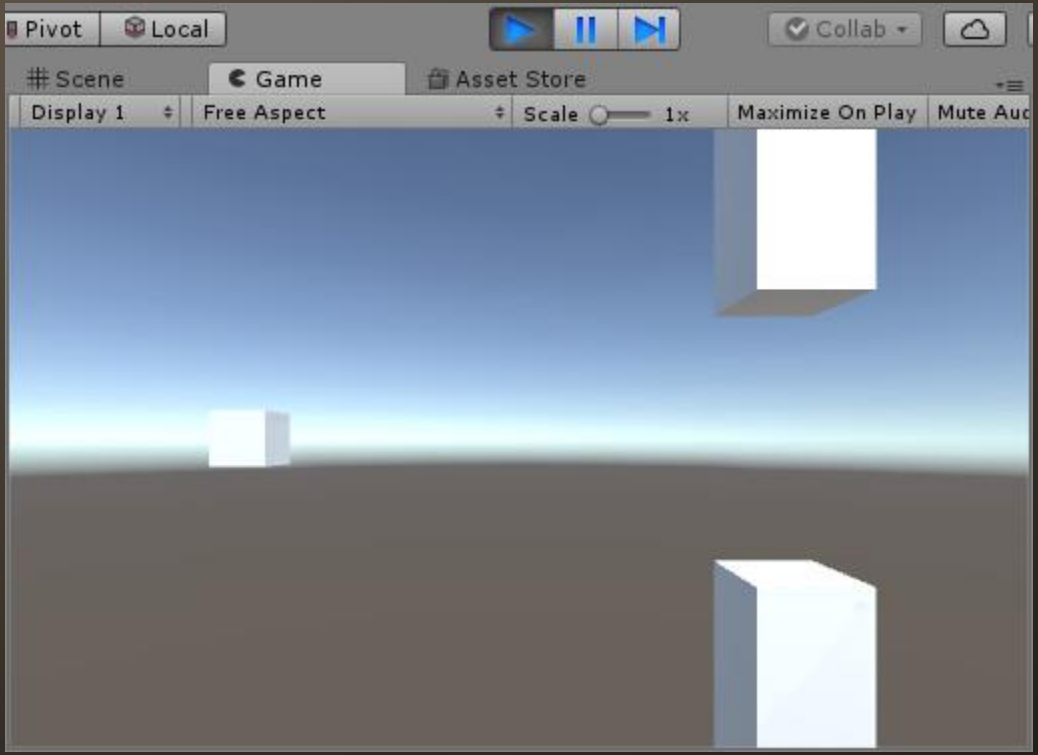




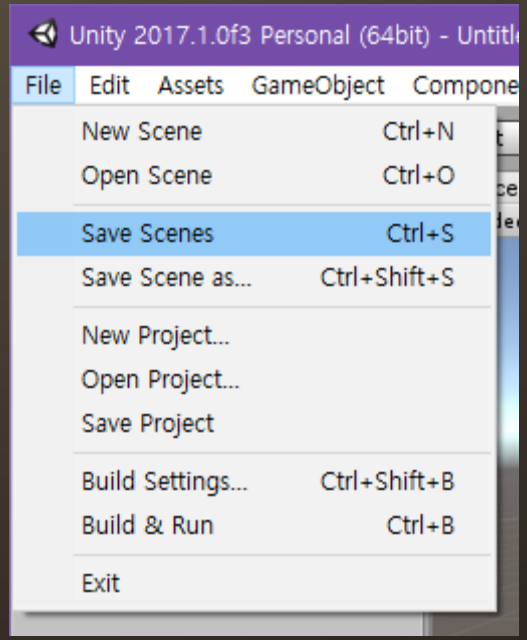
플레이어와 벽의 위치 조정



플레이

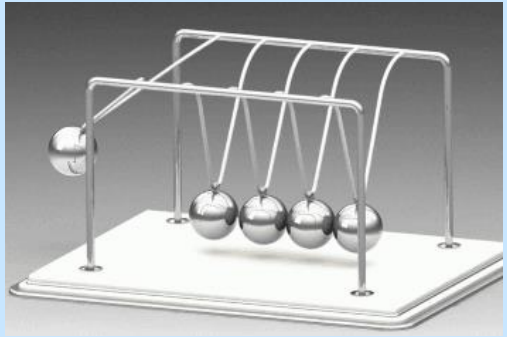

Pivot
Local
Collab
Scene
Game
Asset Store
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Aud

장면 저장



Menu: File > Save Scenes (Ctrl+S)

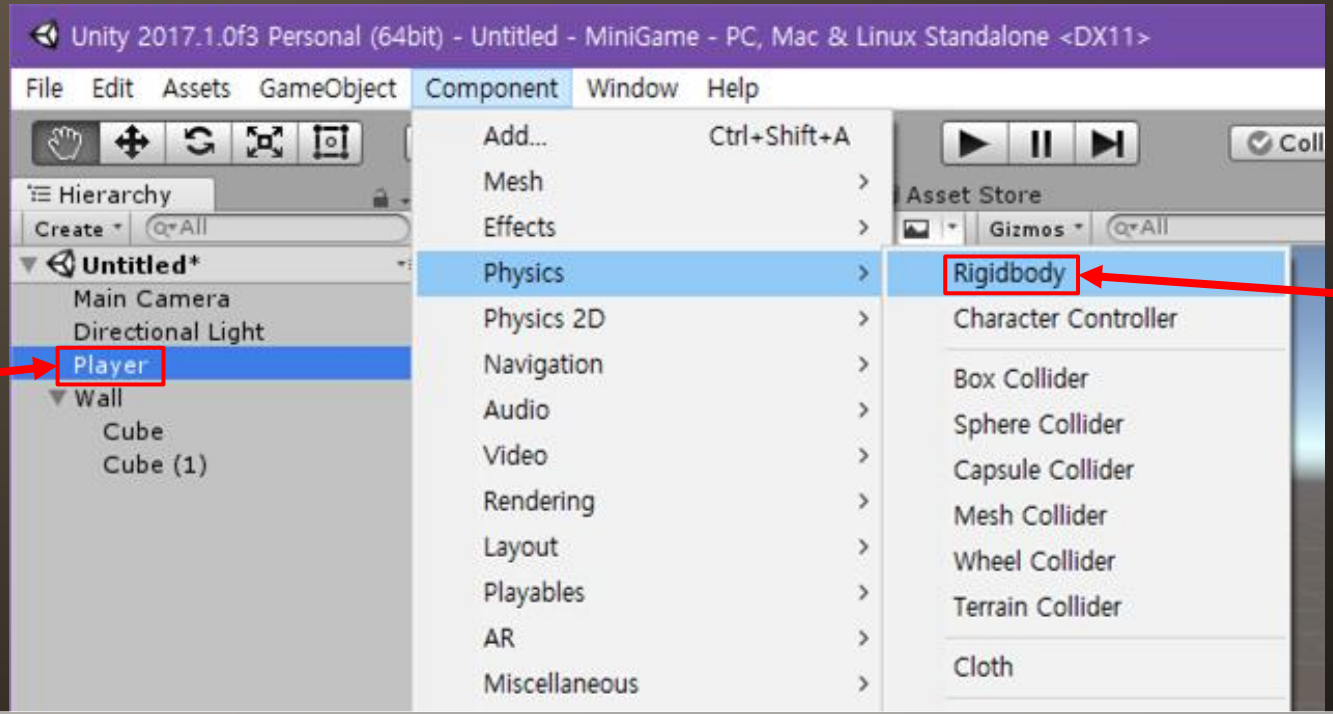
물리 현상



DELICIOUS GAMES

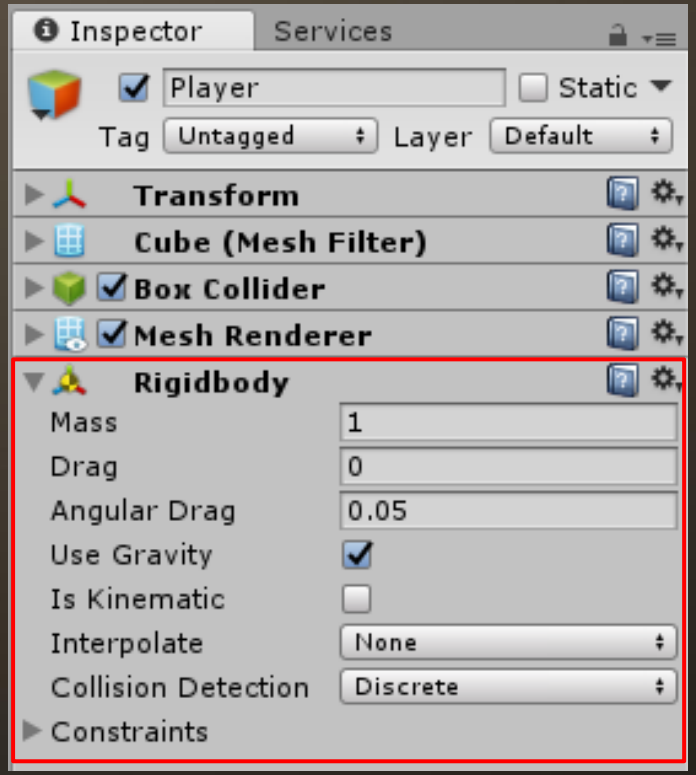
Physics

물체에 물리적 강체 요소 추가

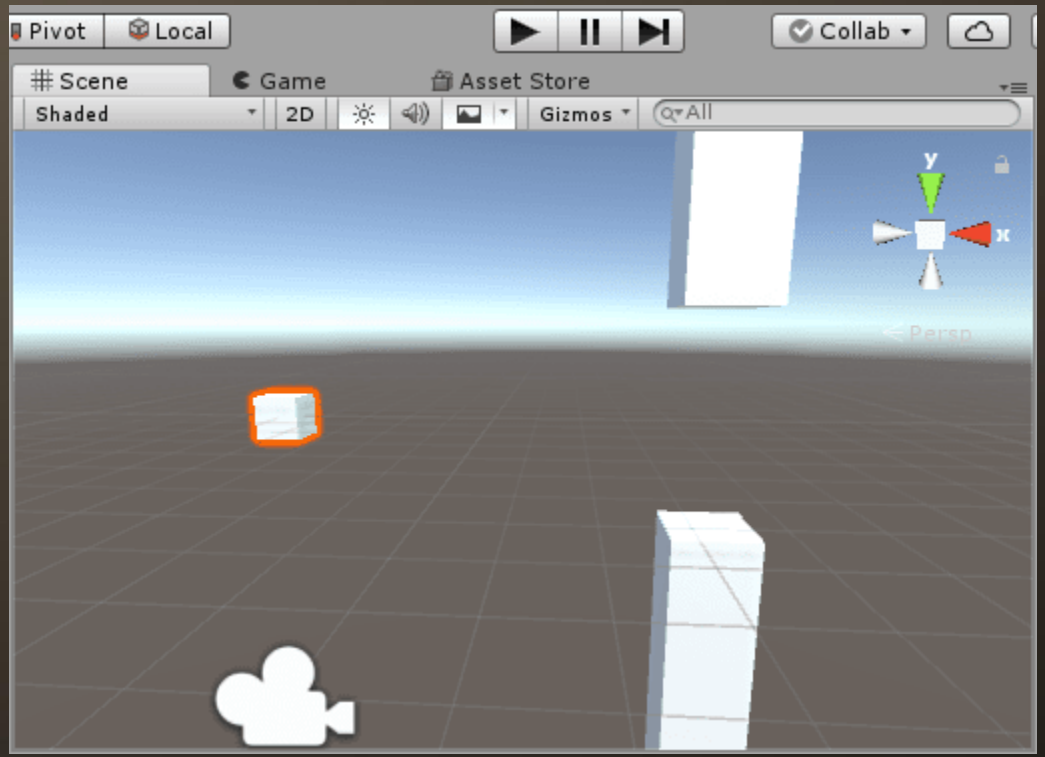


Menu: Component > Physics > Rigidbody

물체에 물리적 강체 요소 추가



플레이


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

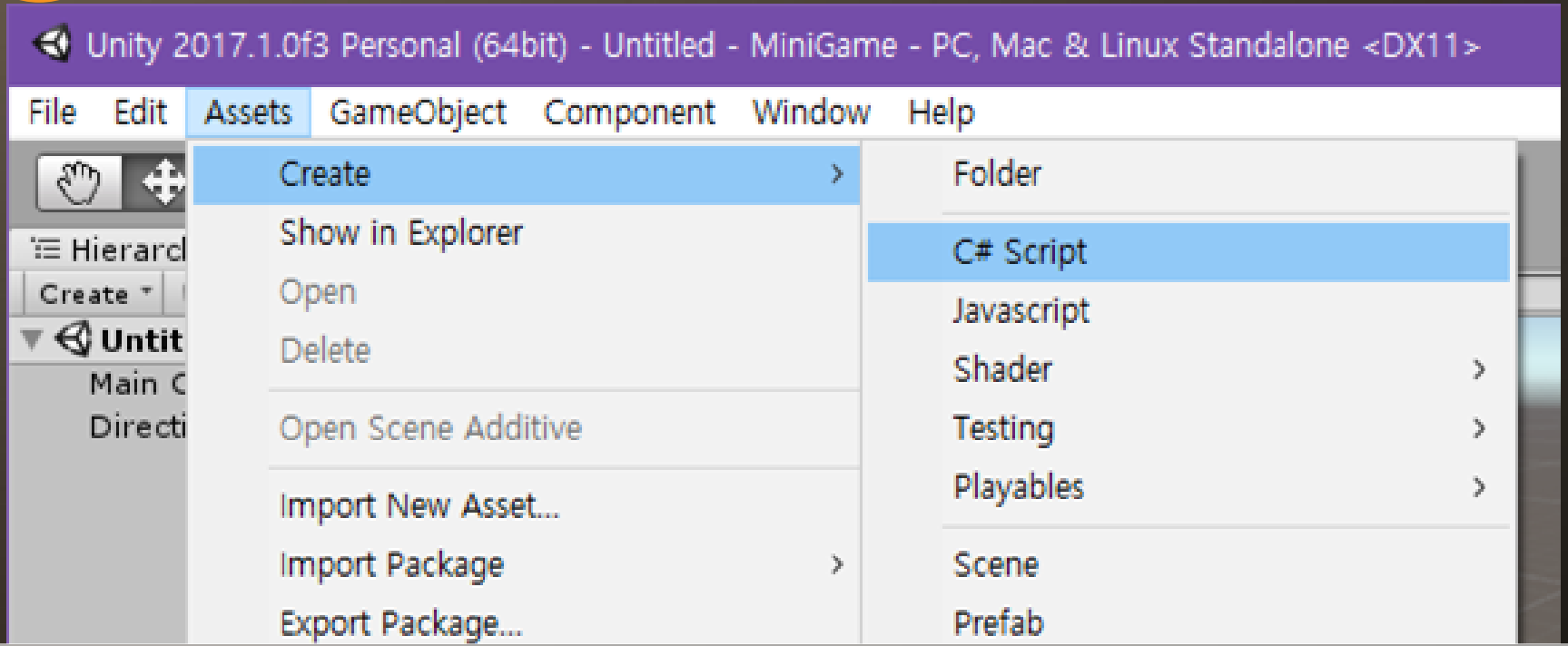
스크립트

DELICIOUS GAMES

Scripts

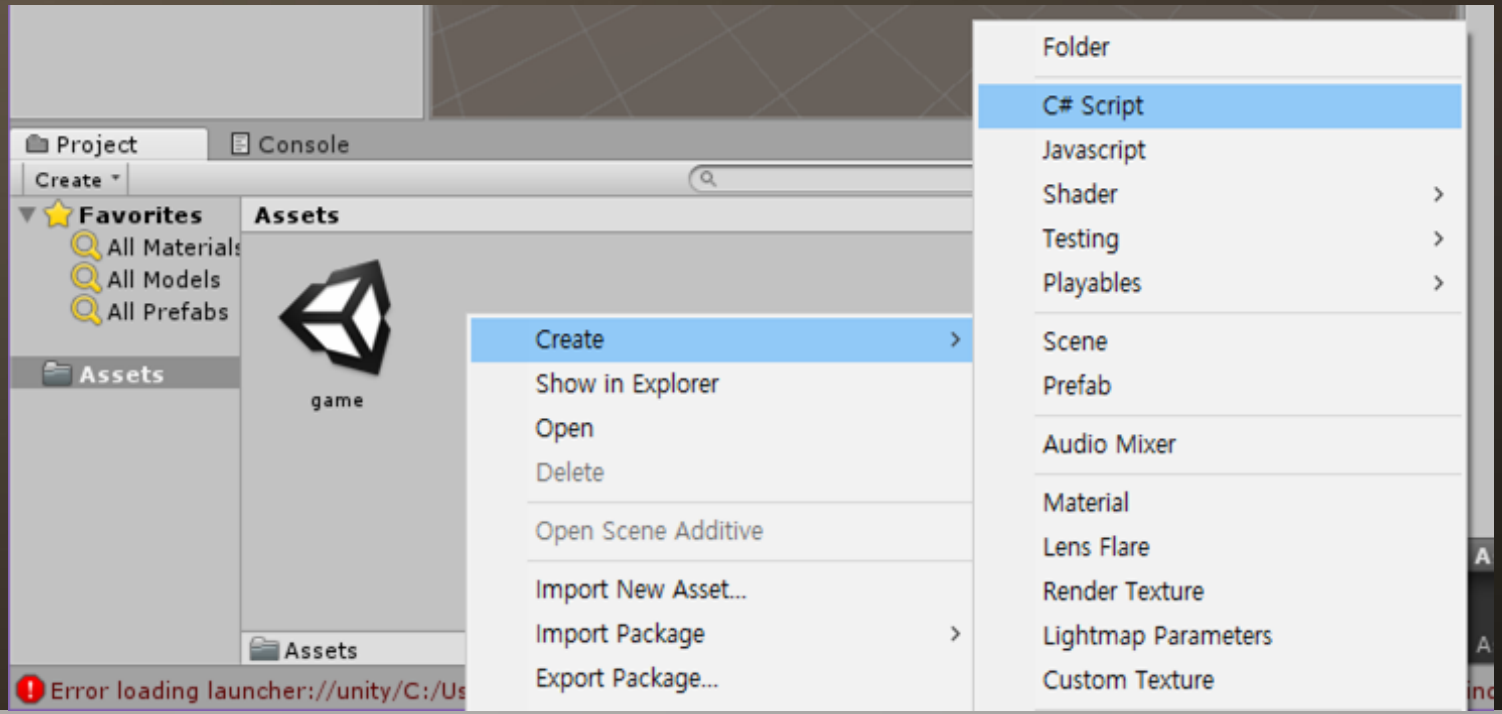
DELICIOUS GAMES

스크립트 추가



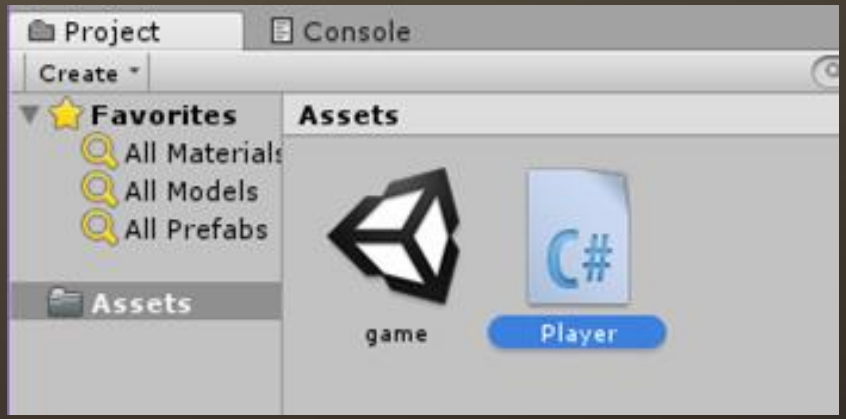
Menu: Assets > Create > C# Script

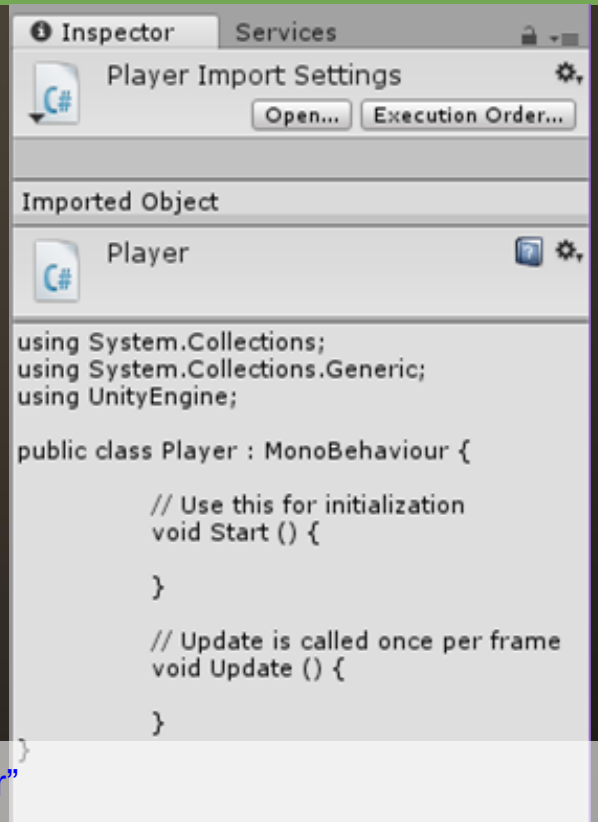
스크립트 추가



Menu: Assets > Create > C# Script

스크립트 추가





Class name: “Player”

DELICIOUS GAMES

스크립트 편집

MiniGame - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

Debug | Any CPU | Unity에 연결

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

솔루션 'MiniGame' (1개 프로젝트)

- MiniGame
  - 참조
  - Assets

Player.cs

MiniGame | Player | Start()

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

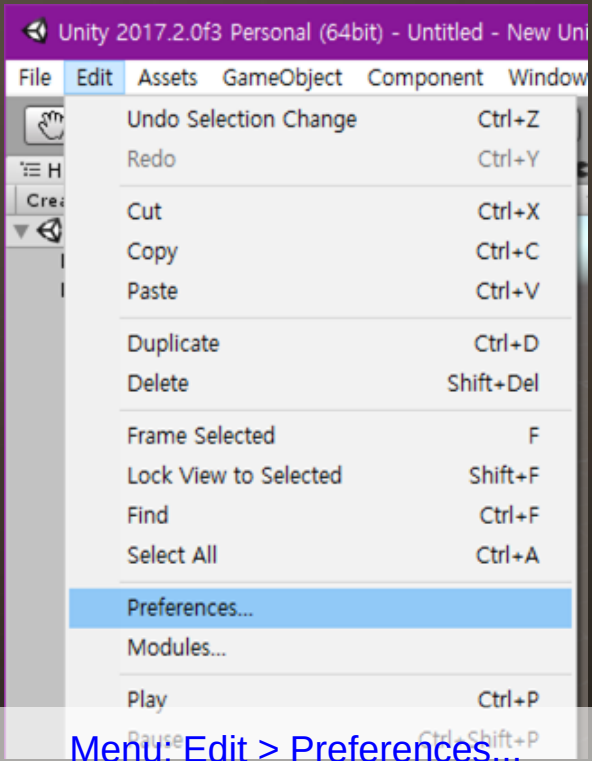
솔루션 탐... 클래스 뷰 속성 관리... 리소스 뷰 팀 탐색기

출력

준비 줄: 1 열: 1 문자: 1 INS 소스 제어에 추가

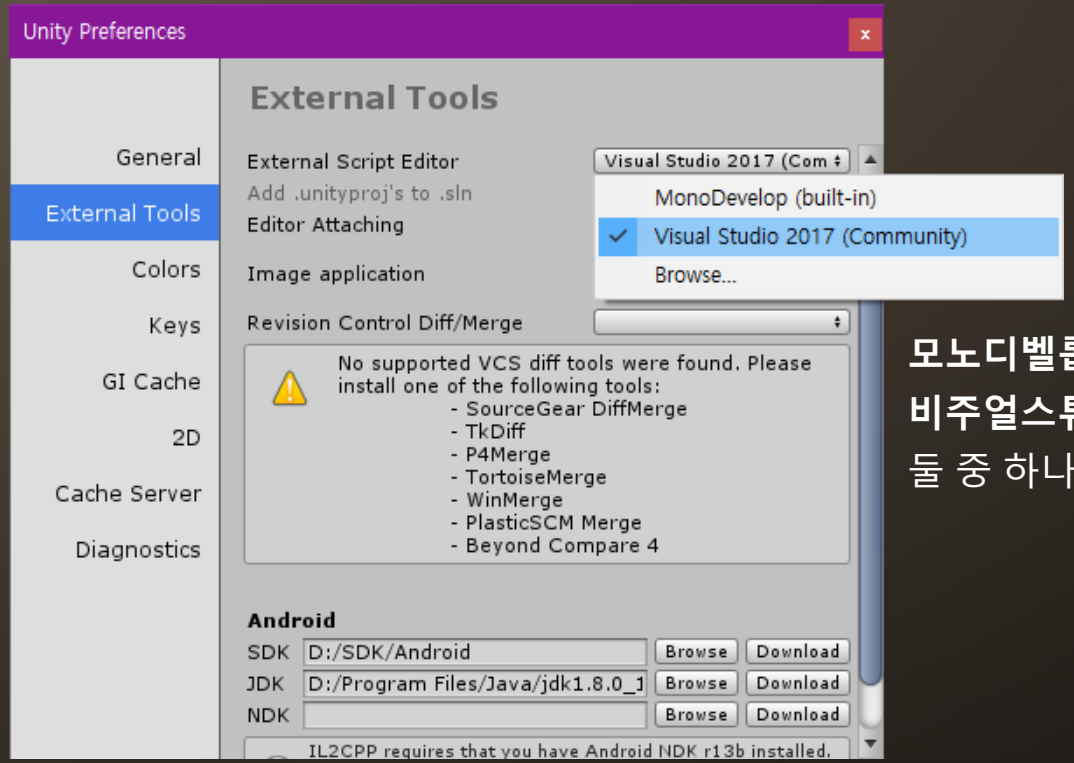
스크립트를 더블 클릭하면 비주얼 스튜디오 실행

스크립트 편집기 선택



Menu: Edit > Preferences...

스크립트 편집기 선택



모노디벨롭과
비주얼스튜디오
둘 중 하나로 선택 가능

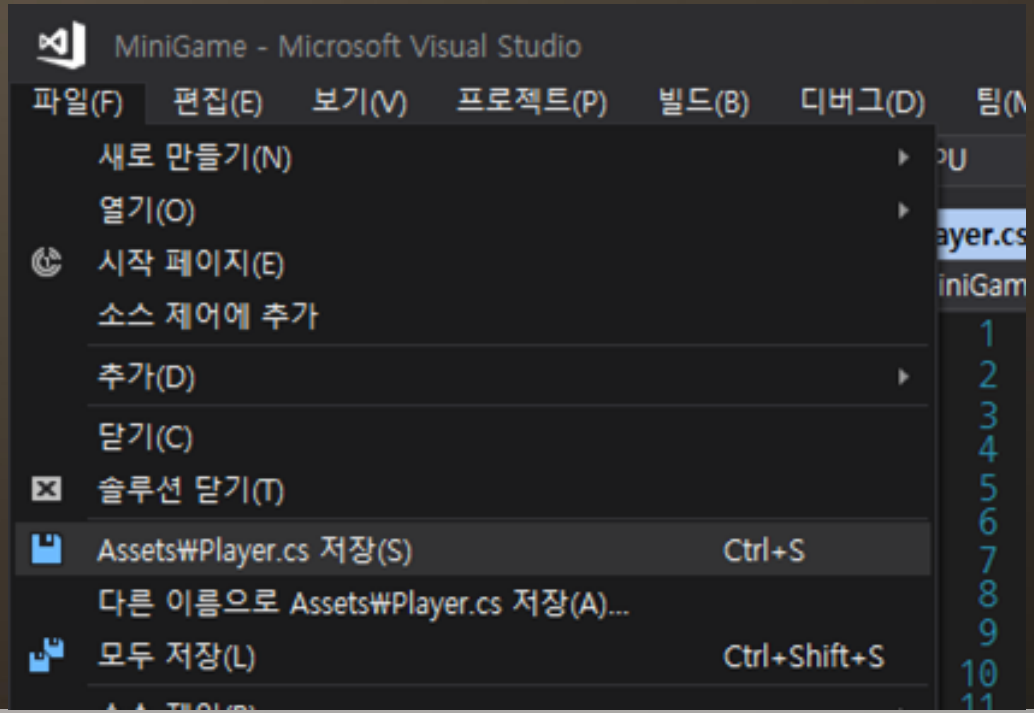
점프 기능 추가

Player.cs

MiniGame | Player | Update()

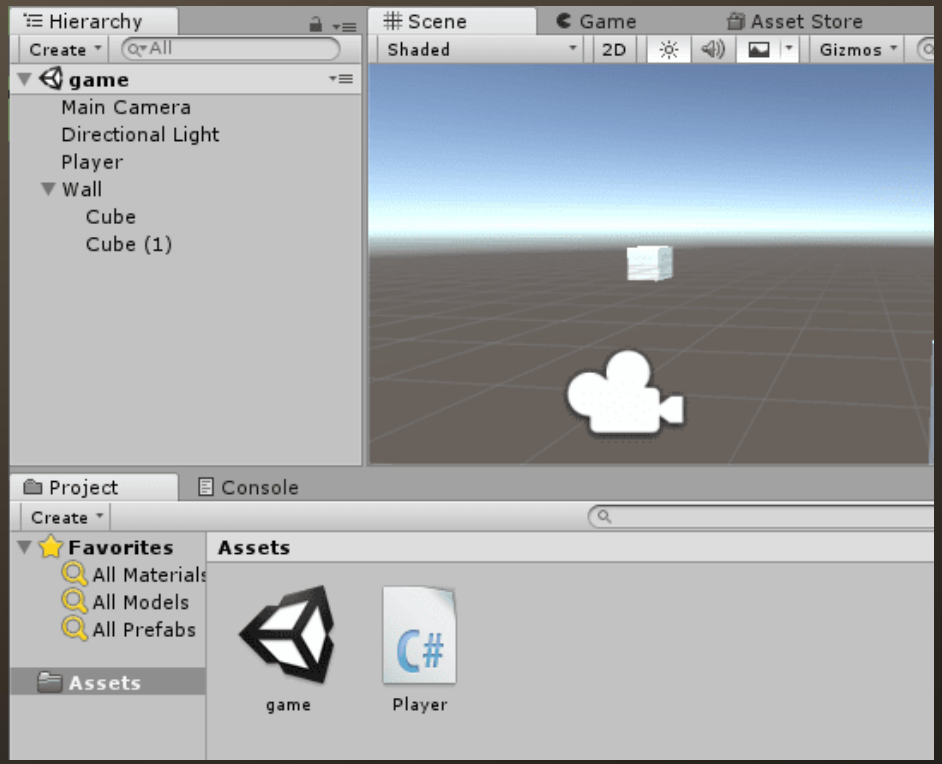
```
public class Player : MonoBehaviour {

        public float jumpPower = 5;

        // Use this for initialization
        void Start () {

        }

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

스크립트 저장

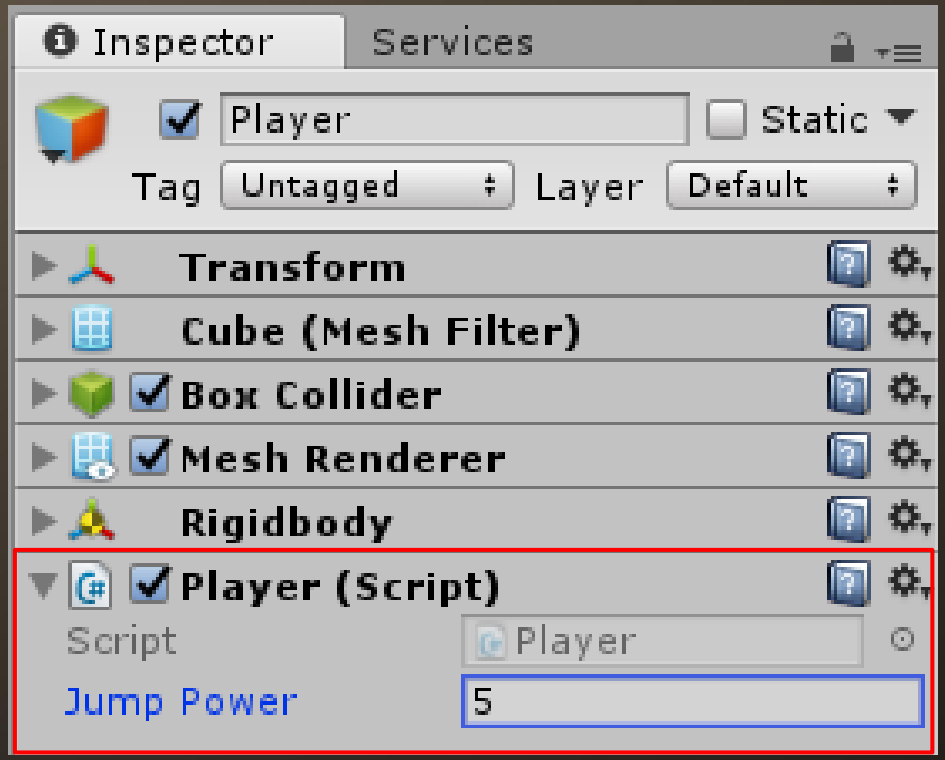


Menu: File > Save (Ctrl+S)

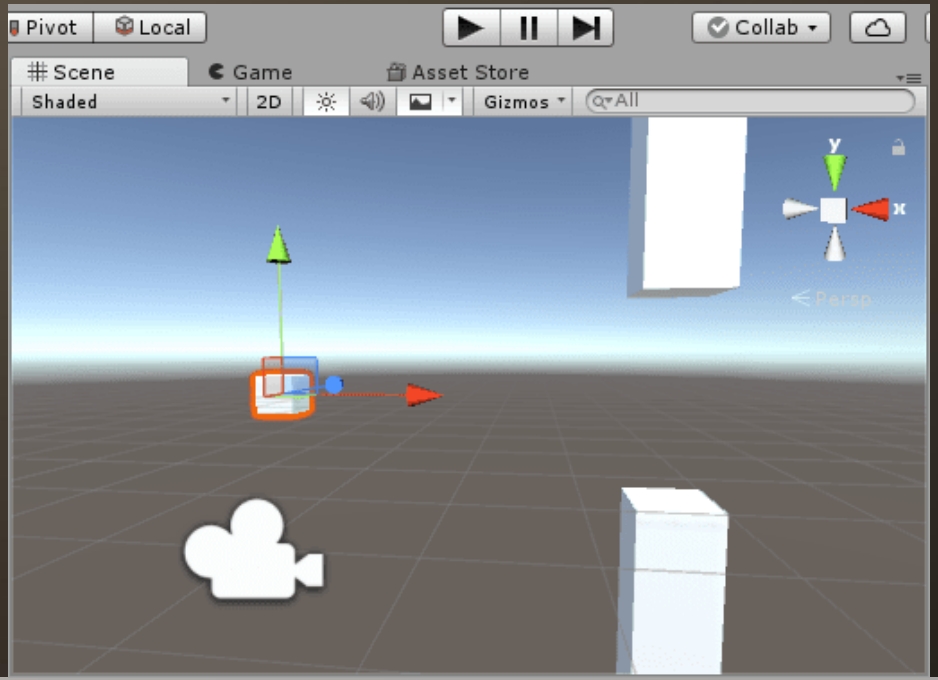
스크립트 연결


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

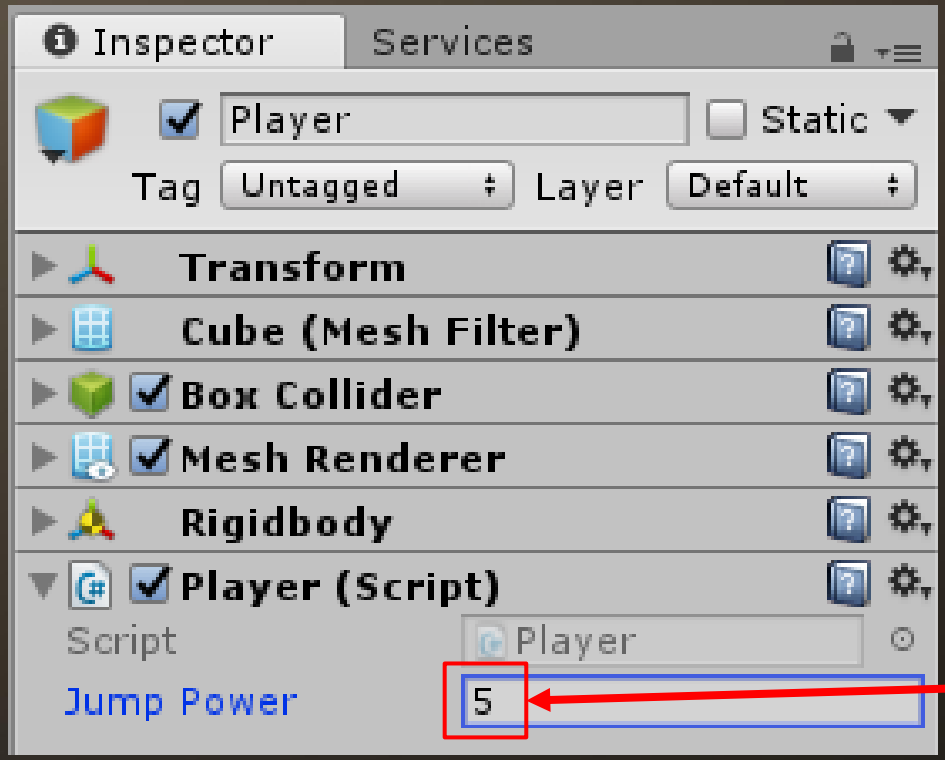

Inspector
Services
Player
Static
Tag
Untagged
Layer
Default
Transform
Cube (Mesh Filter)
Box Collider
Mesh Renderer
Rigidbody
Player (Script)
Script
Player
Jump Power
5

플레이



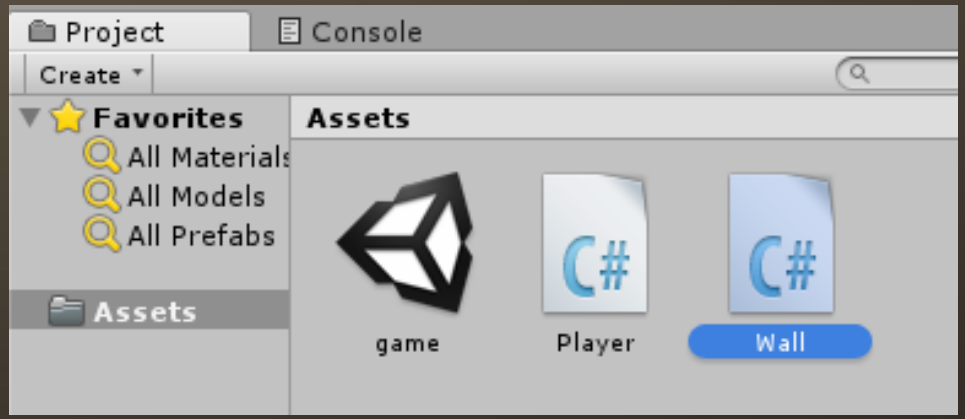
Space 키로 점프 확인

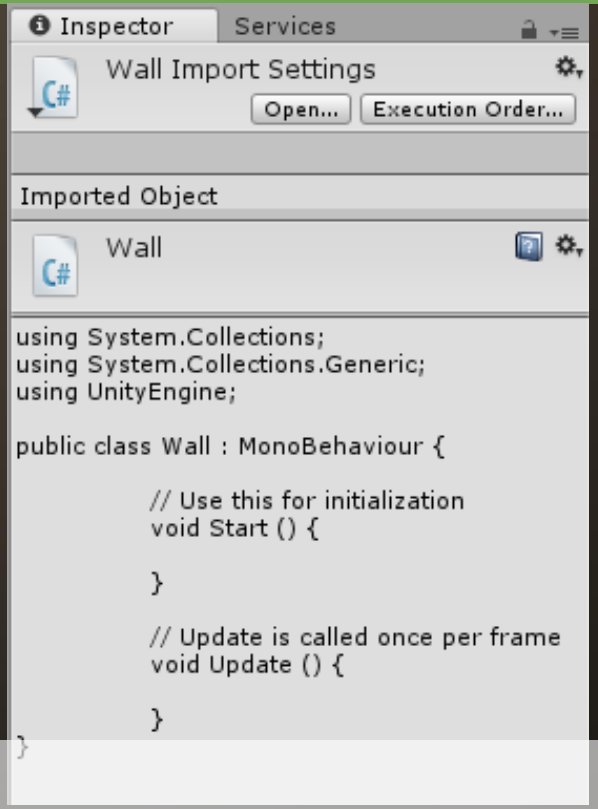
스크립트 연결



수치를 바꿔가며 확인

스크립트 추가





Class name: “Wall”

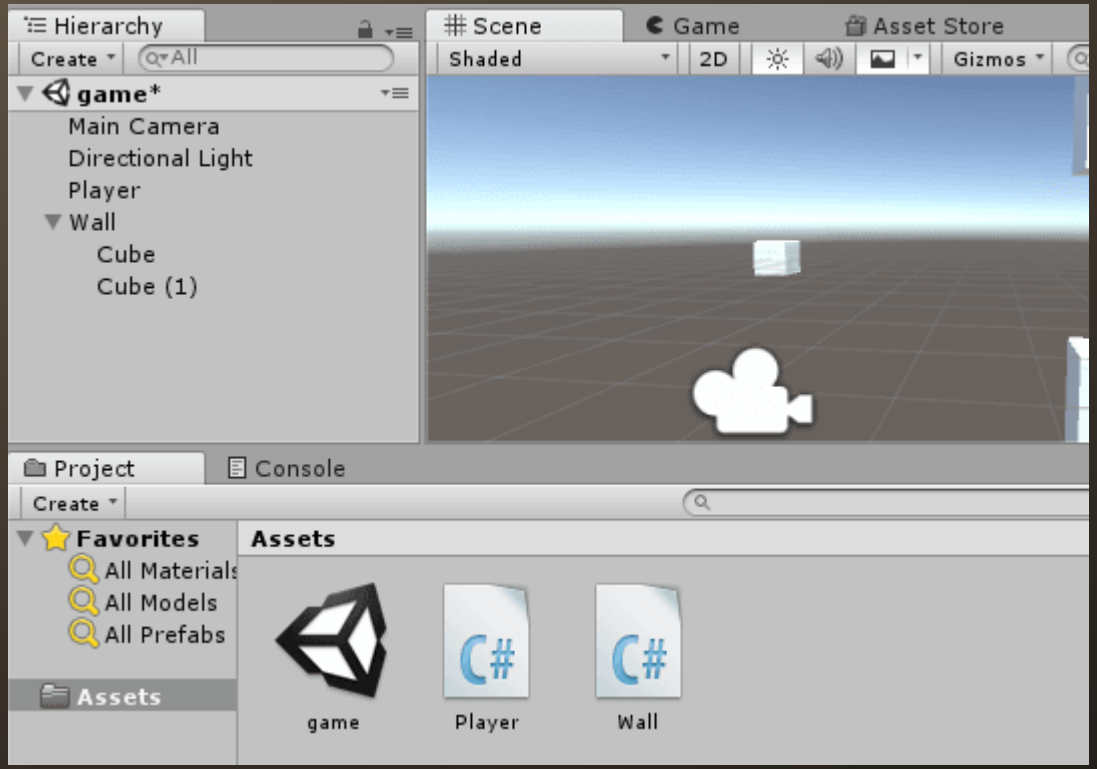
벽 이동 추가

Wall.cs

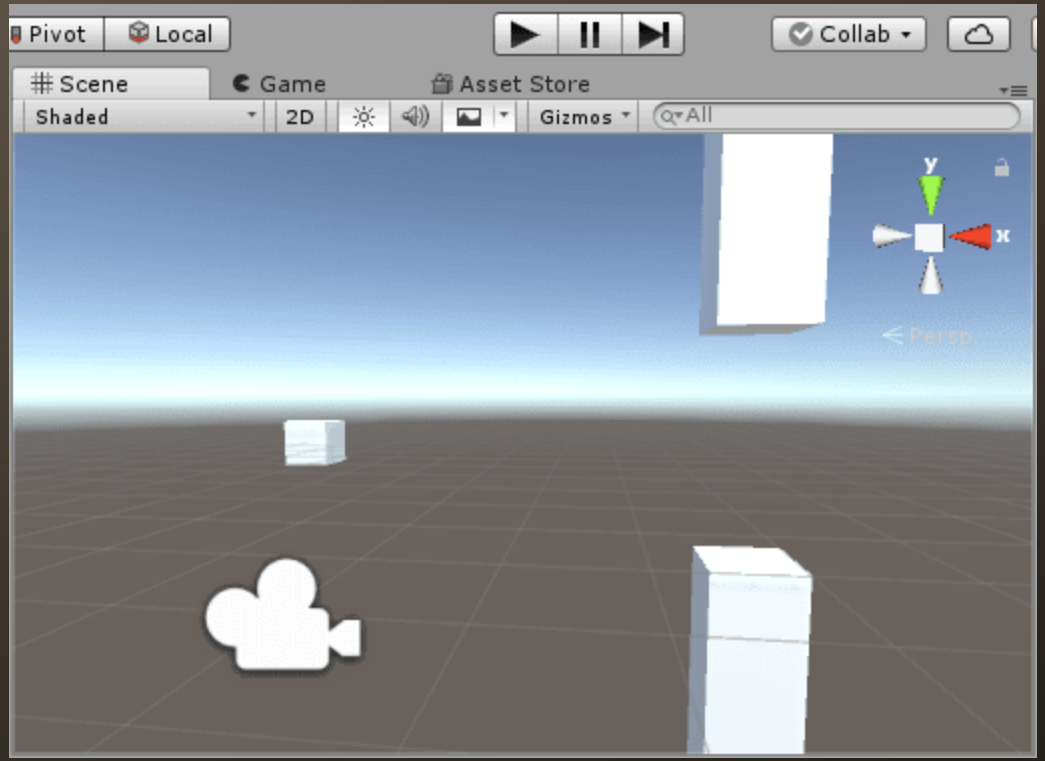
MiniGame | Wall | Update()

```
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);
        }
}
```

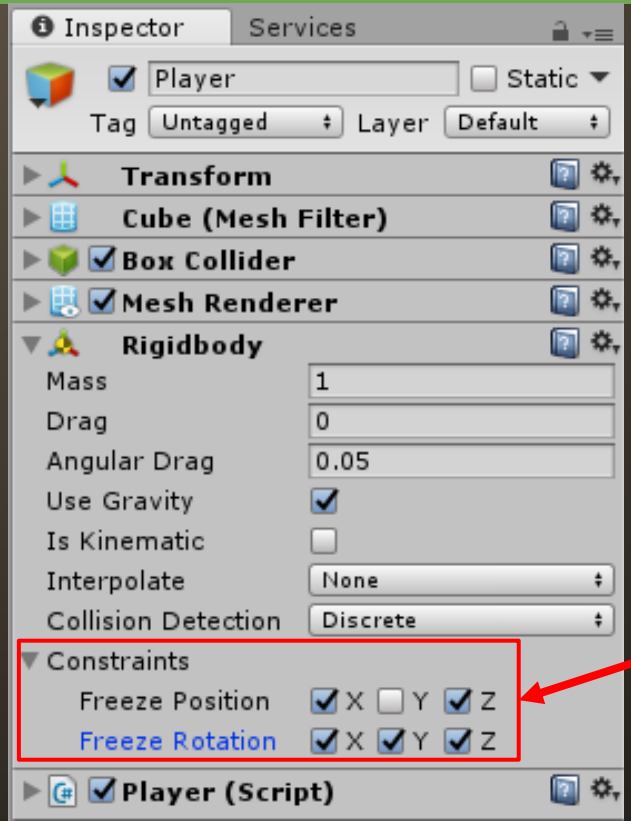
스크립트 연결


Hierarchy
Create
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
Wall

플레이


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

플레이어 큐브 고정시키기

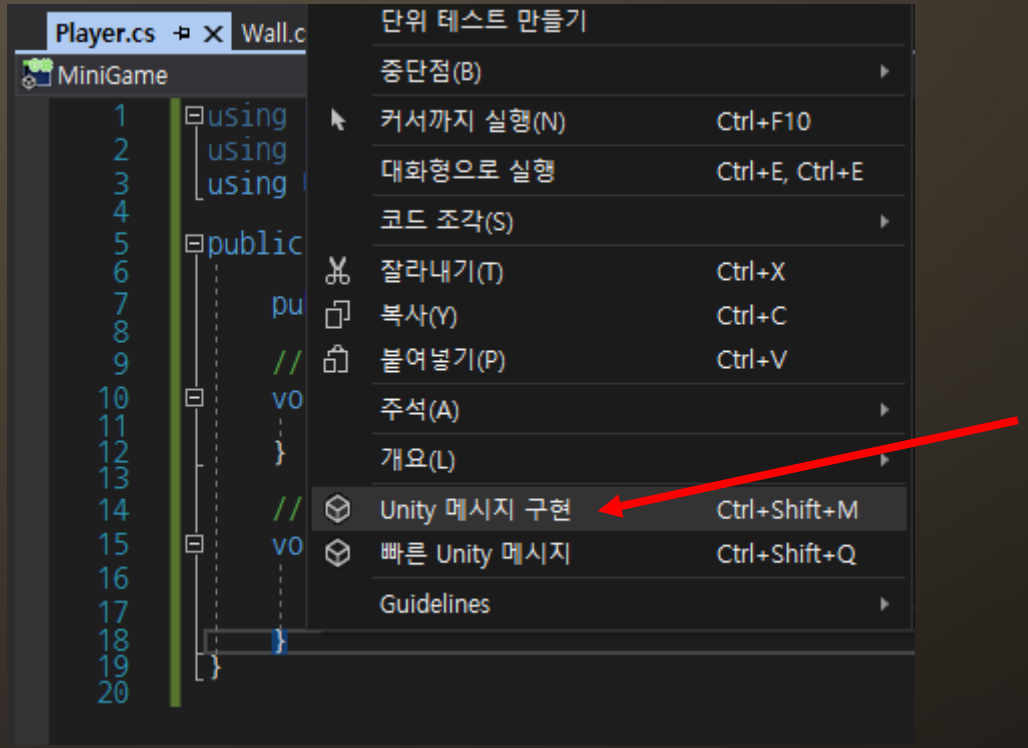


Y축 위치 이동을 제외하고
전부 변화가 없도록 고정

충돌 처리 추가

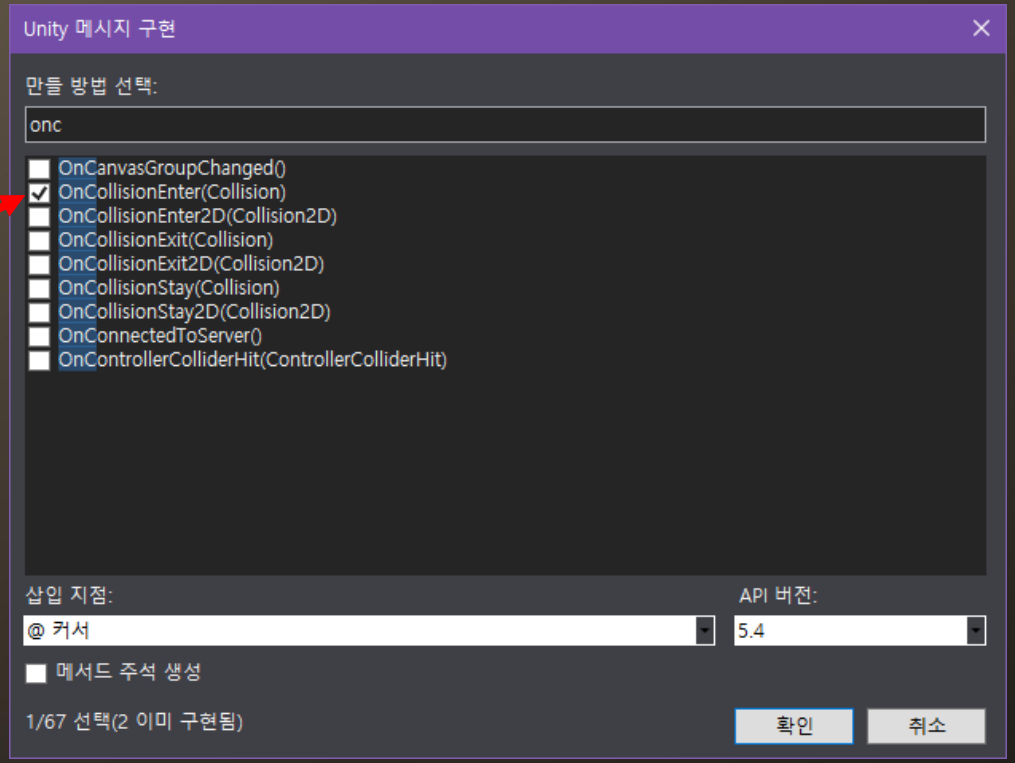
- Player.cs 스크립트 수정
  - 플레이어가 벽에 충돌하면 처음부터 다시 시작하기
  - Visual Studio 편집 화면에서 오른쪽 클릭
  - Unity 메세지 구현 메뉴를 실행

충돌 처리 추가


Player.cs
Wall.c
MiniGame
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

```
5   public class Player : MonoBehaviour {
6
7       public float jumpPower = 5;
8
9       // Use this for initialization
10      void Start () {
11
12      }
13
14      // Update is called once per frame
15      void Update () {
16          if (Input.GetButtonDown("Jump"))
17              GetComponent<Rigidbody>().velocity = new Vector3(0, jumpPo
18      }
19
20      private void OnCollisionEnter(Collision collision)
21      {
22
23      }
24  }
25
```

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

새로 로딩 추가

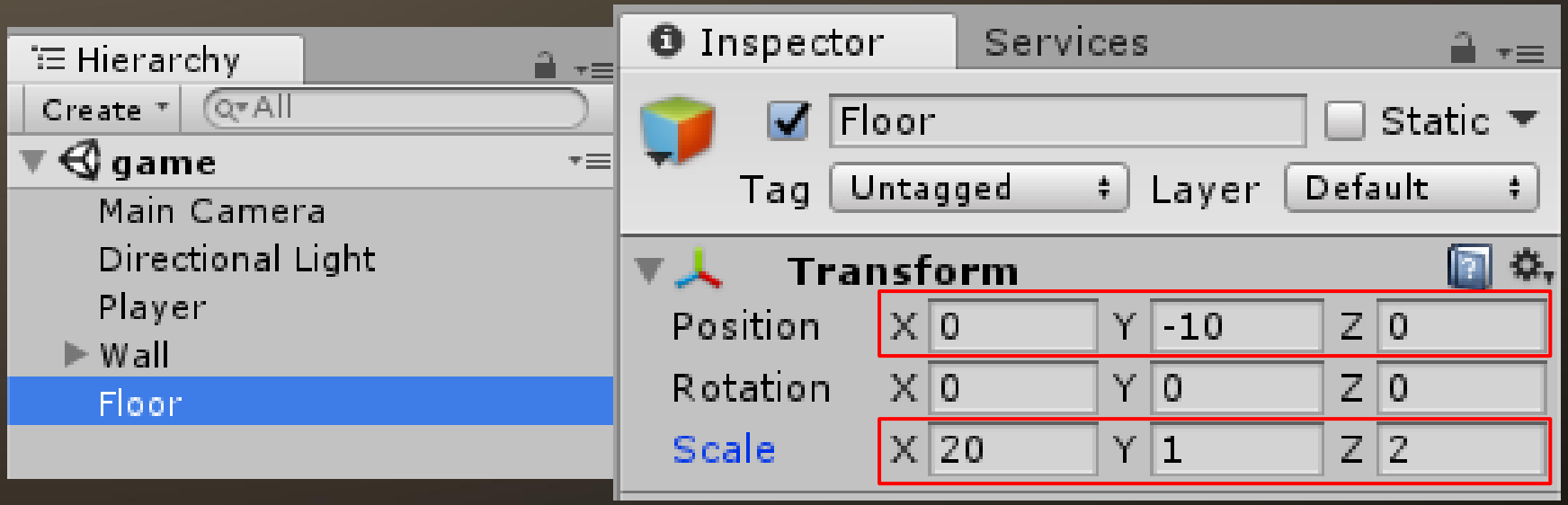
Player.cs

MiniGame | Player | OnCollisionEnter(Collision col

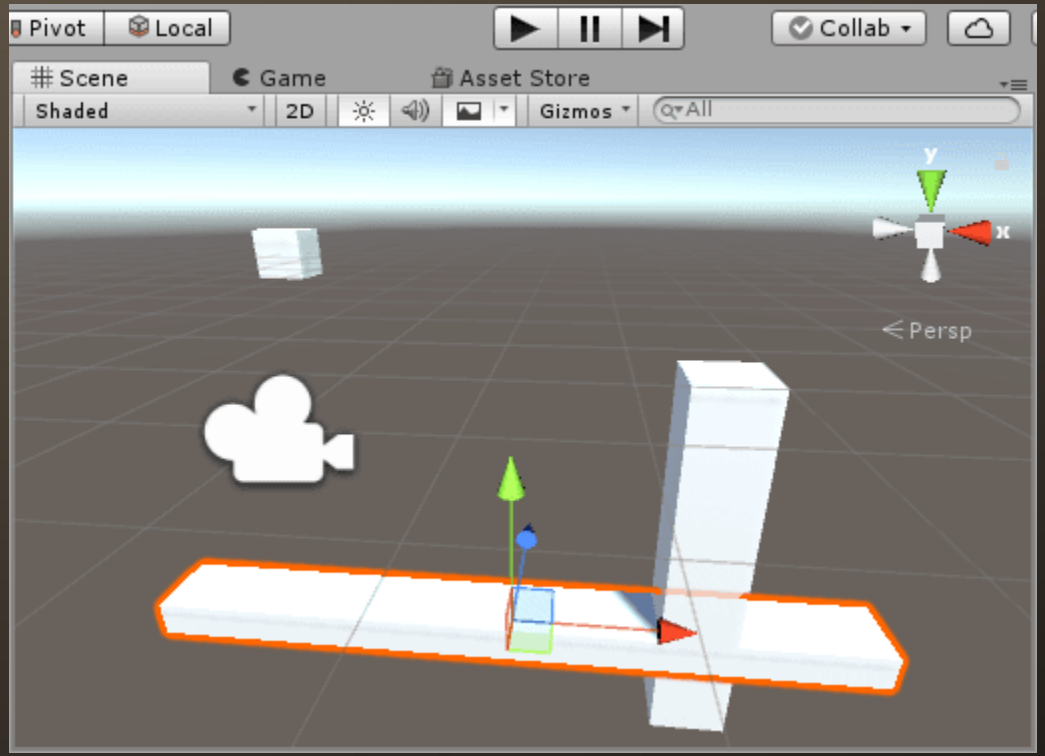
```
private void OnCollisionEnter(Collision collision)
{

SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```

바닥 추가

6. GameObject > 3D Object > Cube



플레이

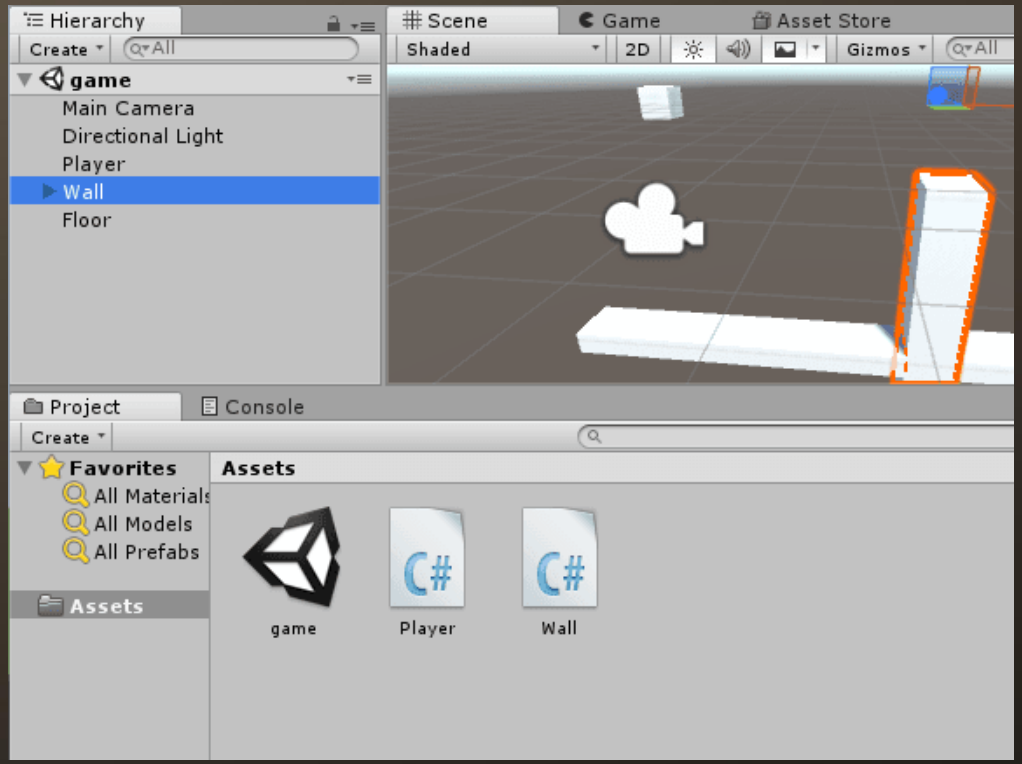

Pivot
Local
Collab
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x
Persp

프리팹

DELICIOUS GAMES

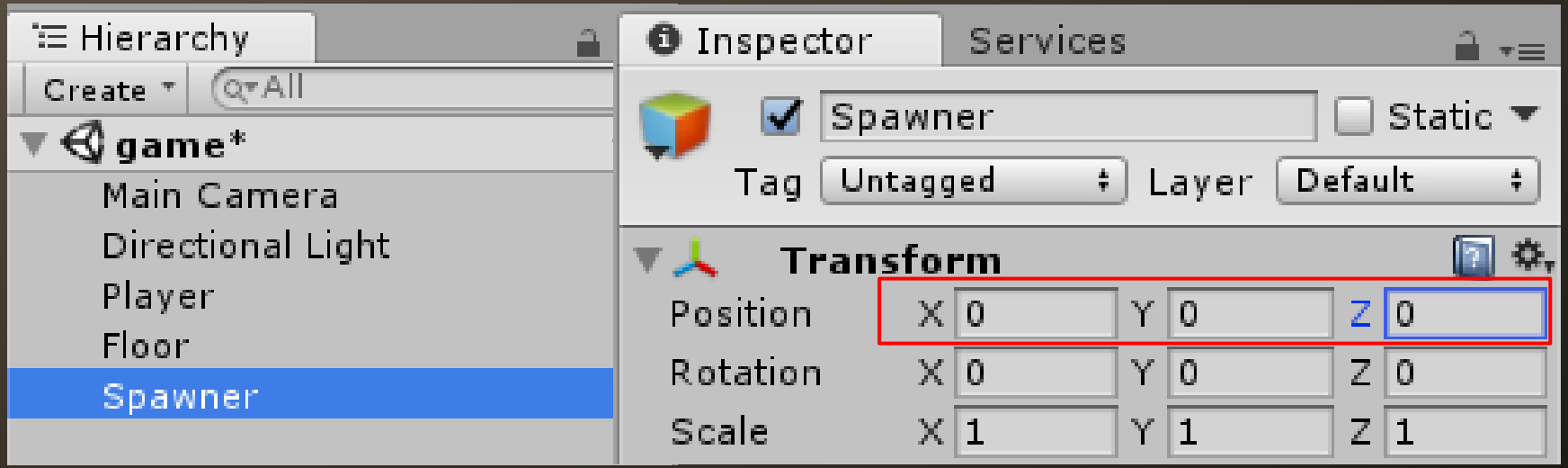
Prefabs

벽 프리팹 만들기

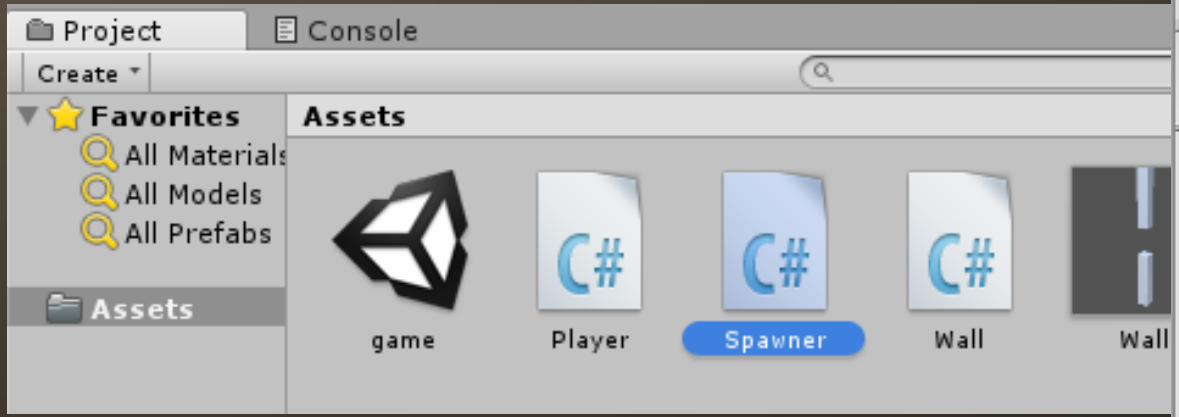

Hierarchy
Create
All
game
Main Camera
Directional Light
Player
Wall
Floor
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Wall

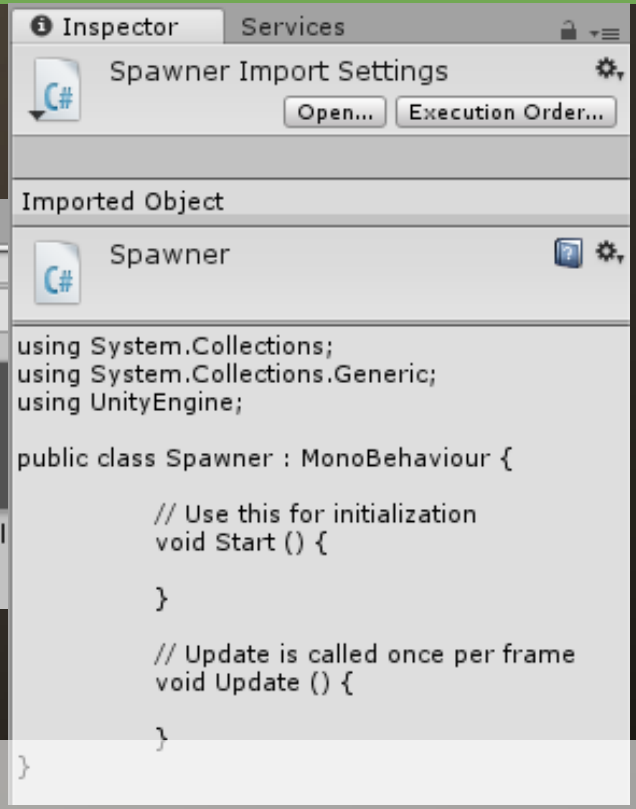
스크립트 실행용 빈 오브젝트 생성

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
    }
```

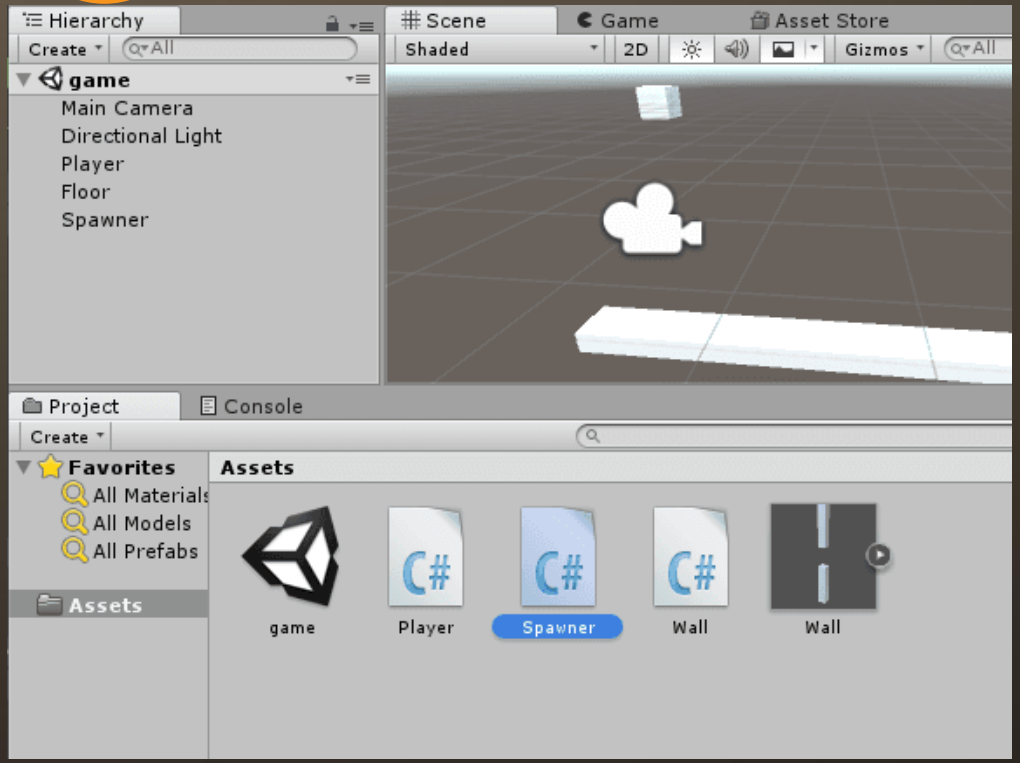
일정 시간마다 프리팹 생성

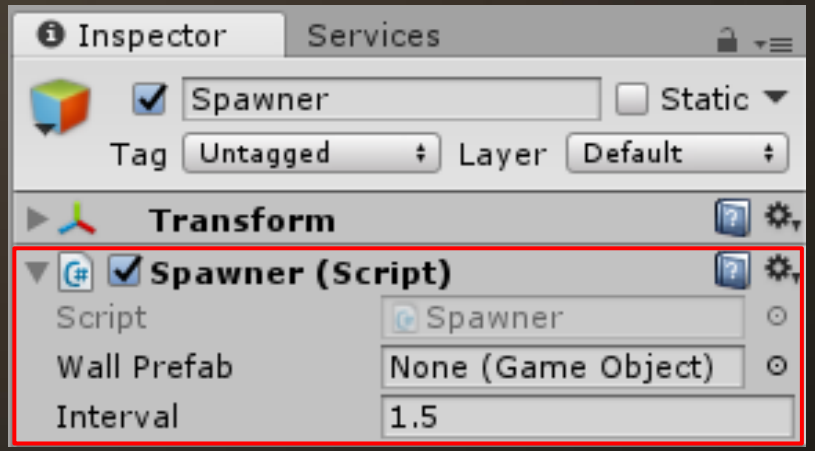
Spawner.cs

MiniGame | Spawner | Update()

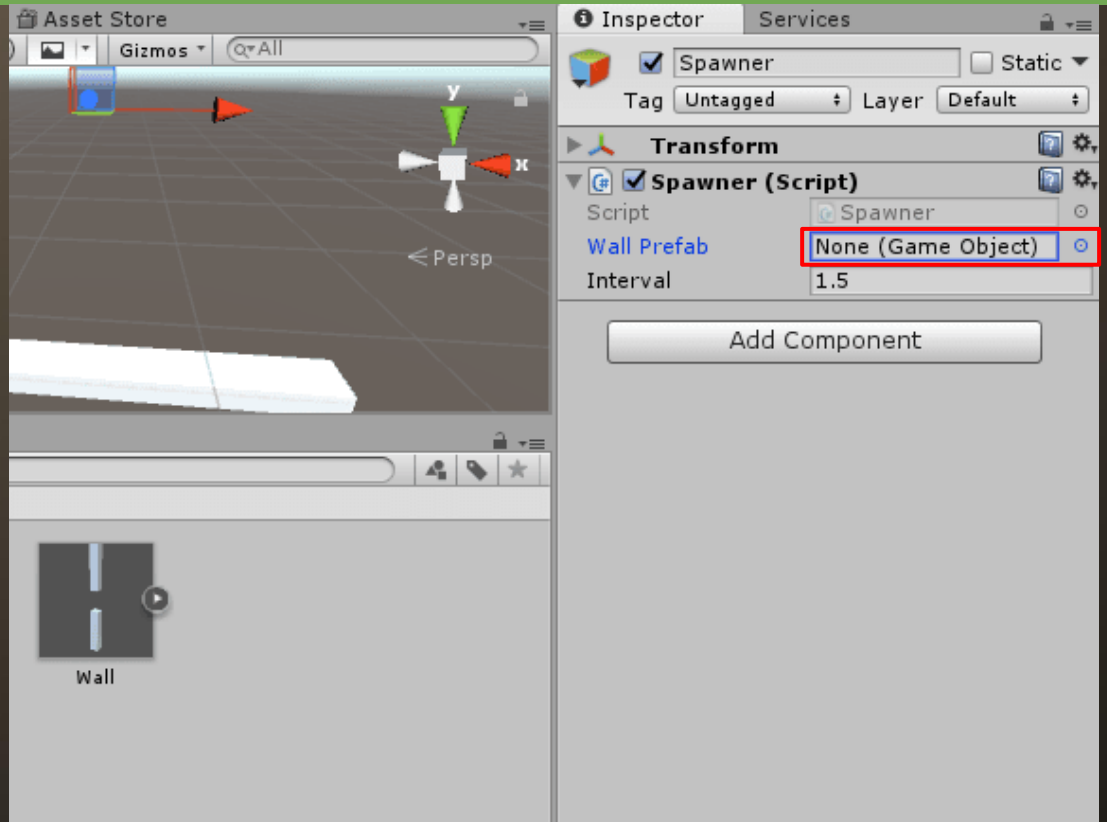
```
        // Update is called once per frame
        void Update () {
            term += Time.deltaTime;
            if (term >= interval)        // 일정 시간이 지나면
            {
                Instantiate(wallPrefab, transform.position
                                        , transform.rotation);
 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

스크립트 연결

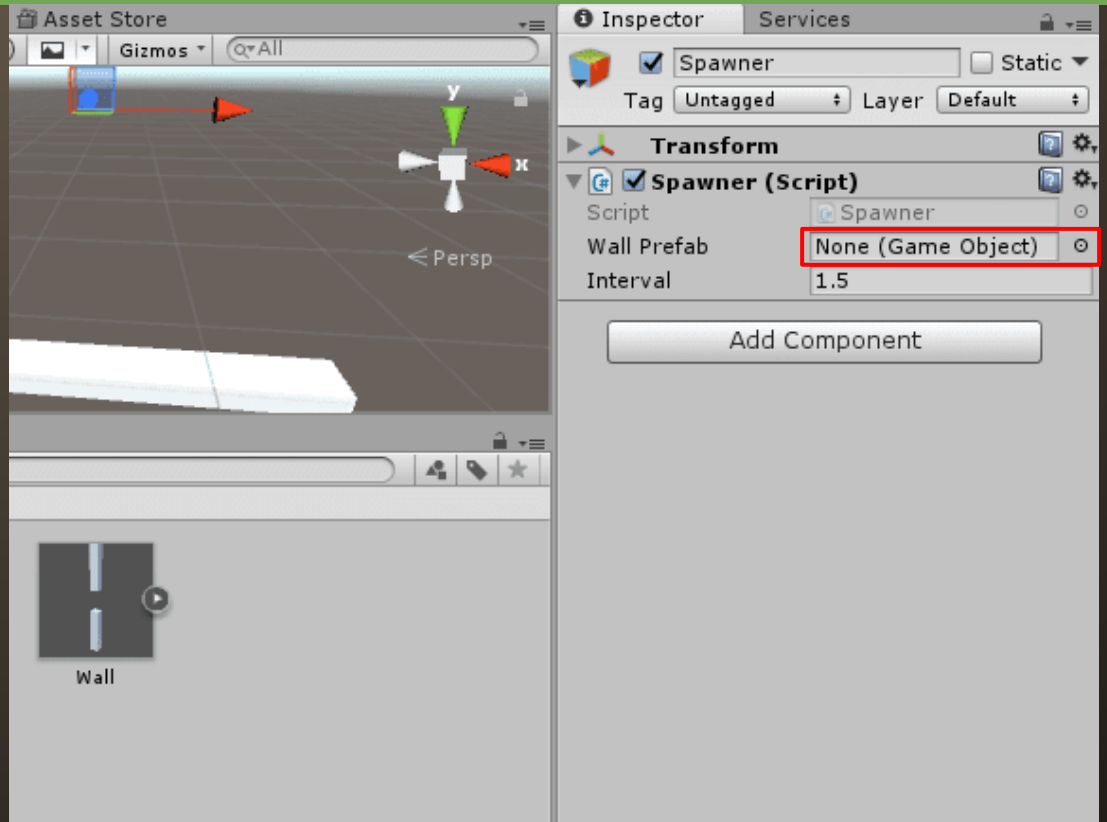

Hierarchy
Create
All
game
Main Camera
Directional Light
Player
Floor
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Spawner
Wall
Wall


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

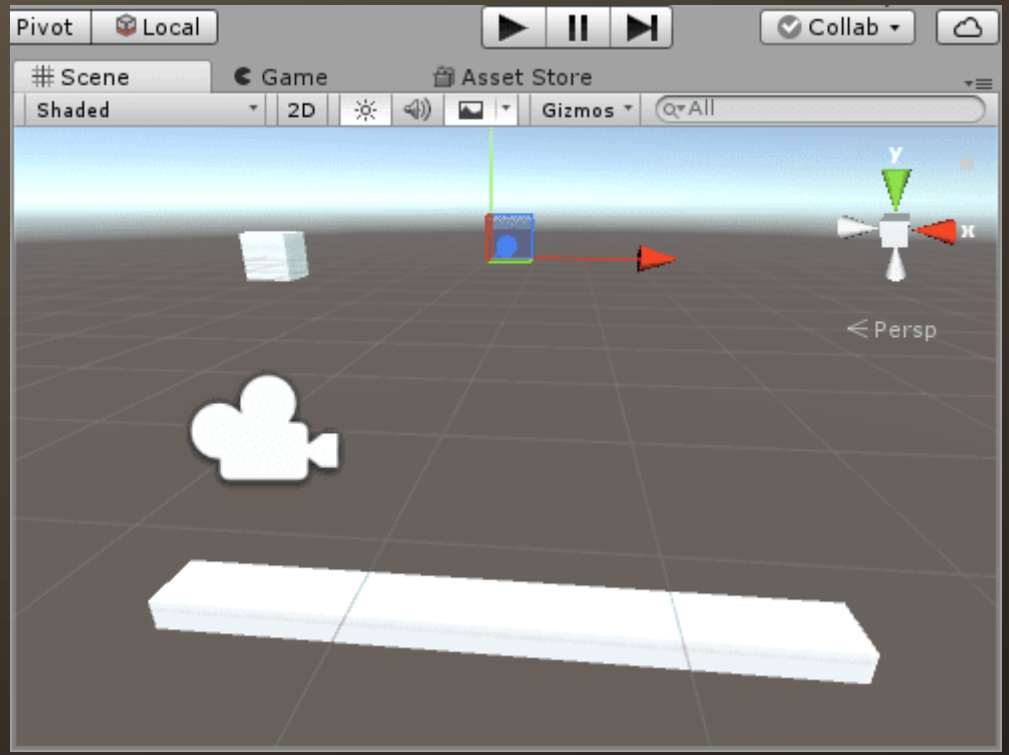
프리팹 연결 (선택 방식)


Asset Store
Gizmos
All
y
x
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
All
y
x
Persp

스폰 위치 조정

Hierarchy

Create | All

game

- Main Camera
- Directional Light
- Player
- Floor
- Spawner

Inspector | Services

Spawner | Static

Tag: Untagged | Layer: Default

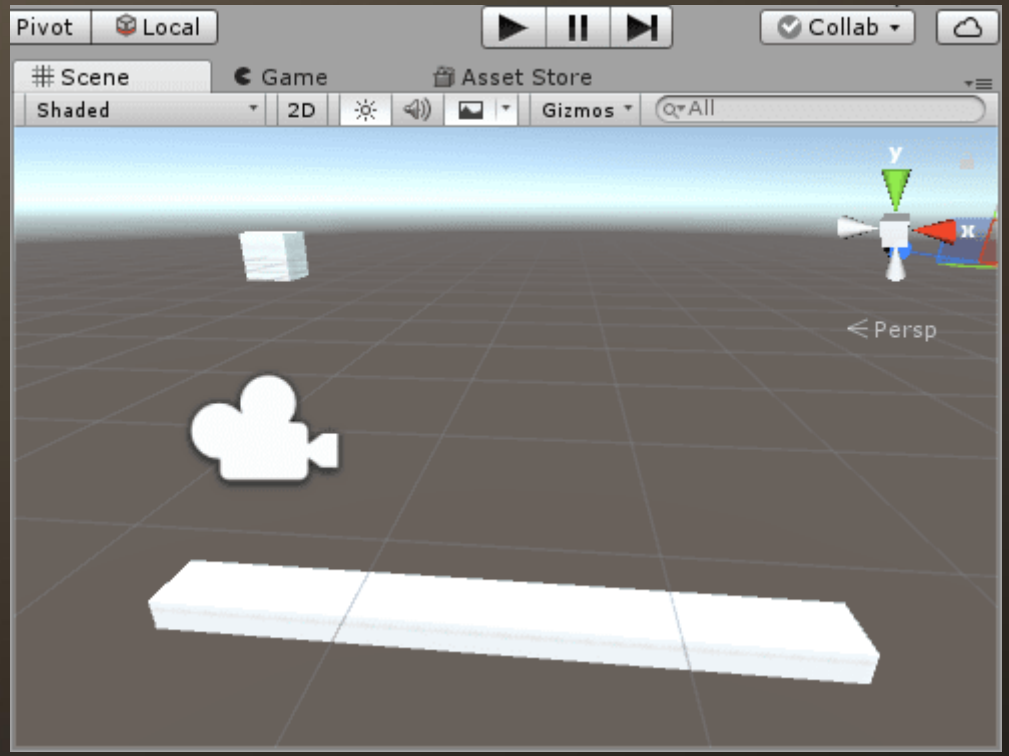
Transform

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
All
y
x
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;
    }
```

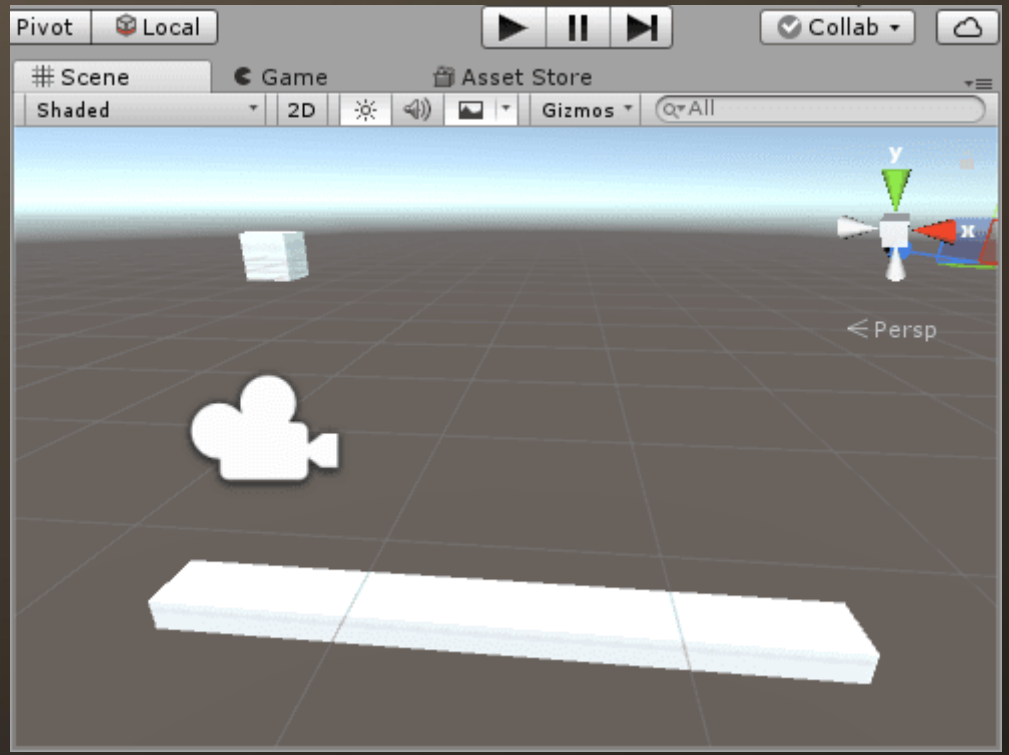
벽의 높이 조정

Spawner.cs

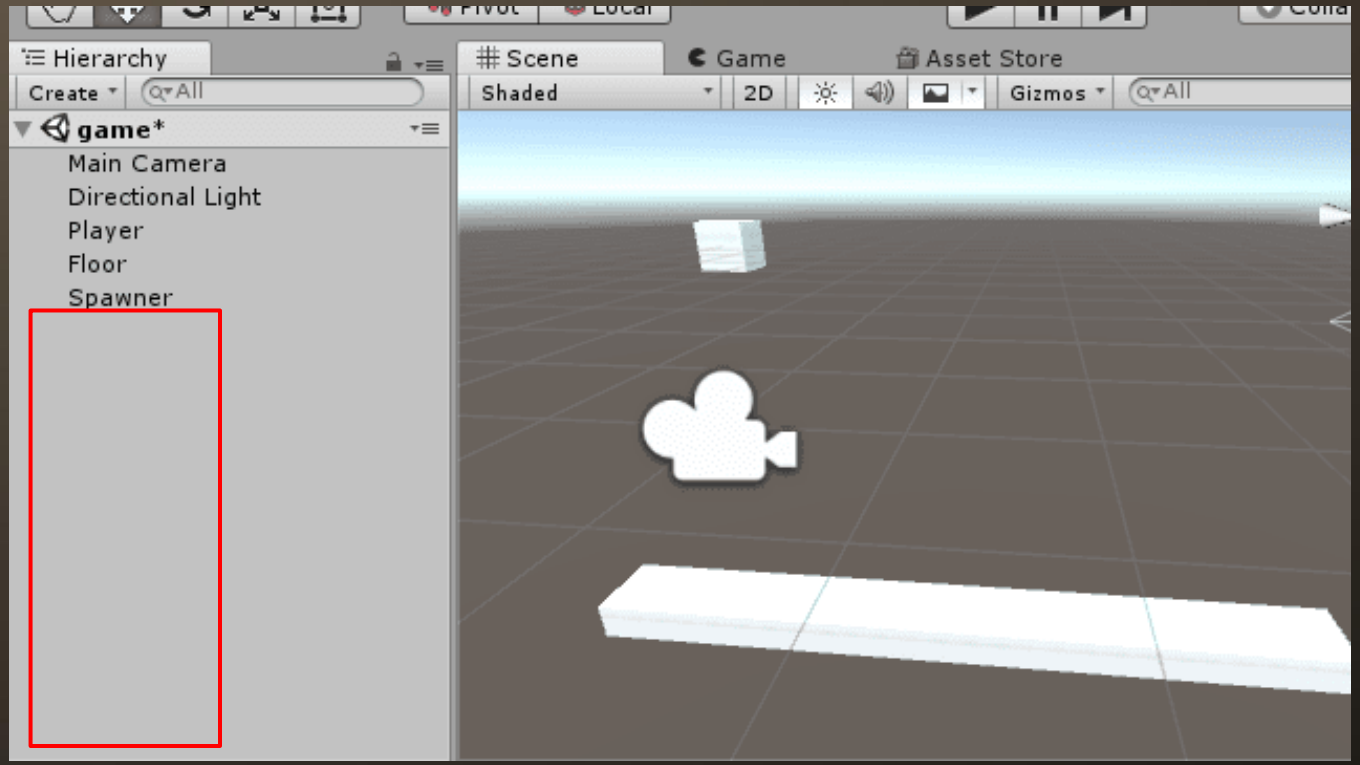
MiniGame | Spawner | Update()

```
        // 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
All
y
x
Persp

플레이


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

DELICIOUS GAMES

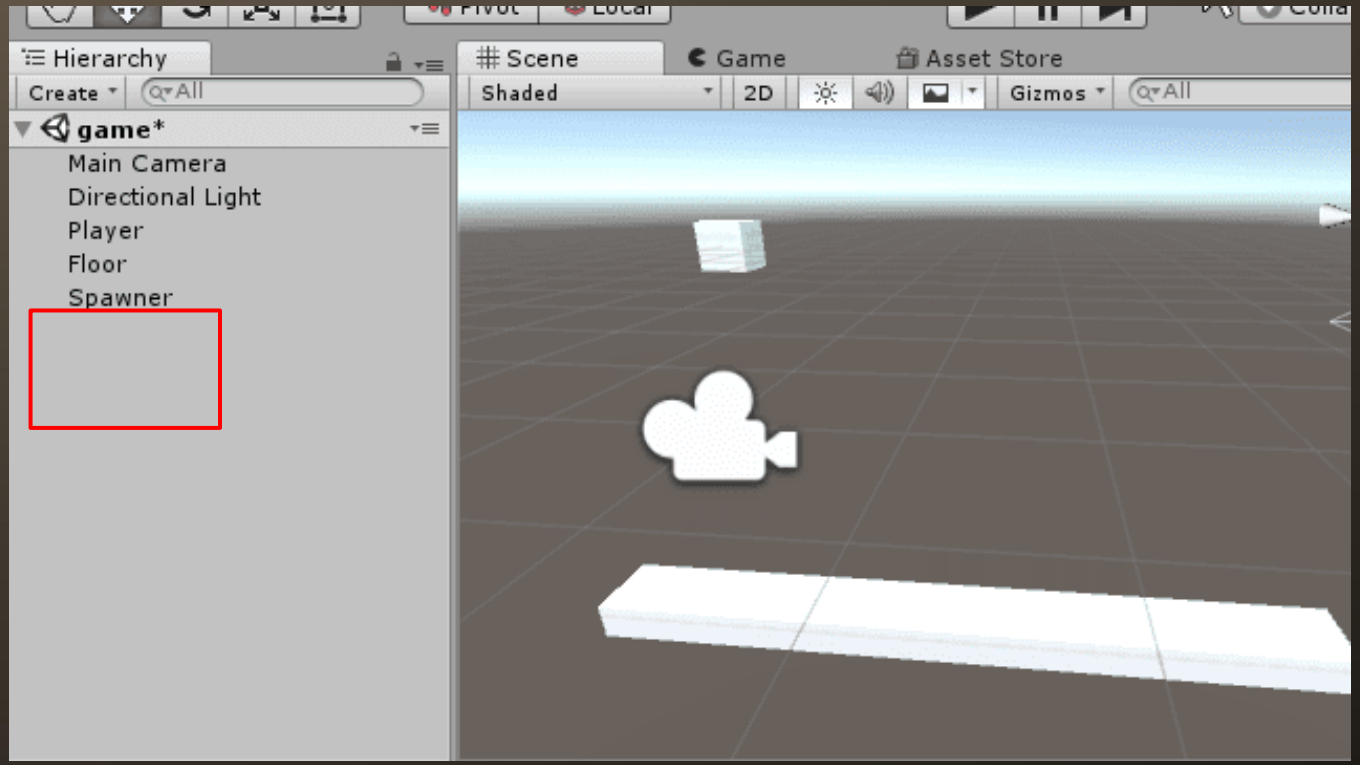
벽이 자동으로 사라지게

Wall.cs

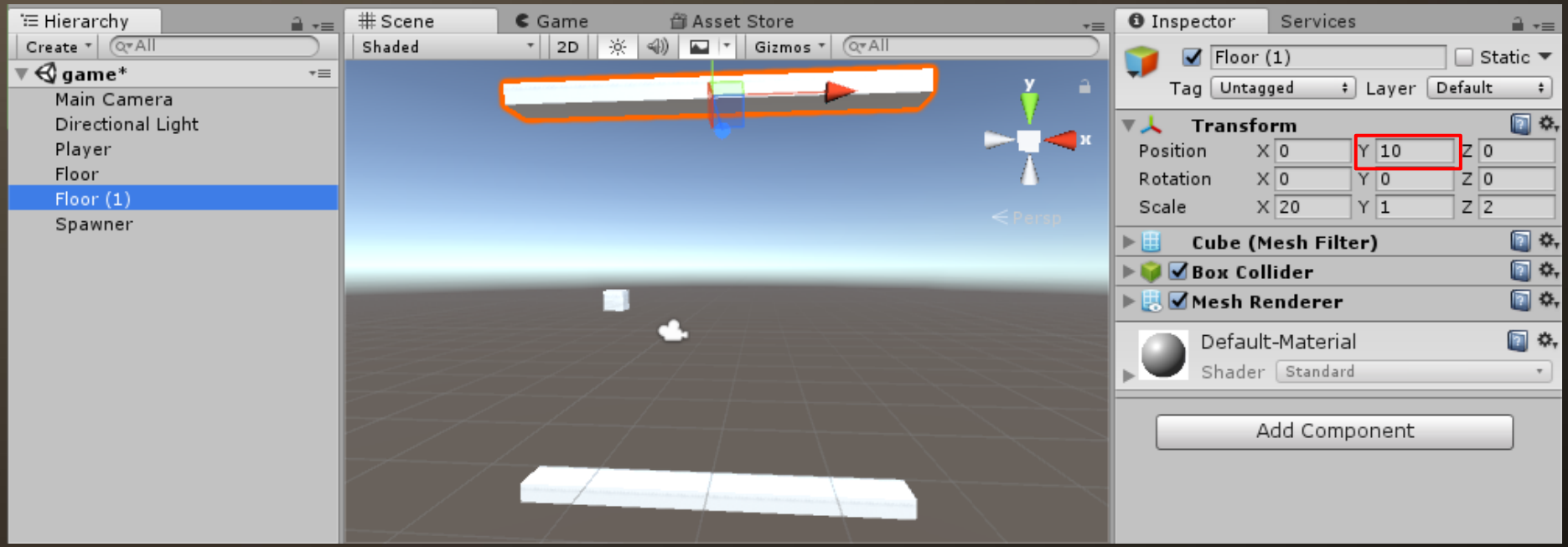
MiniGame | Wall | Update()

```
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);
        }
}
```

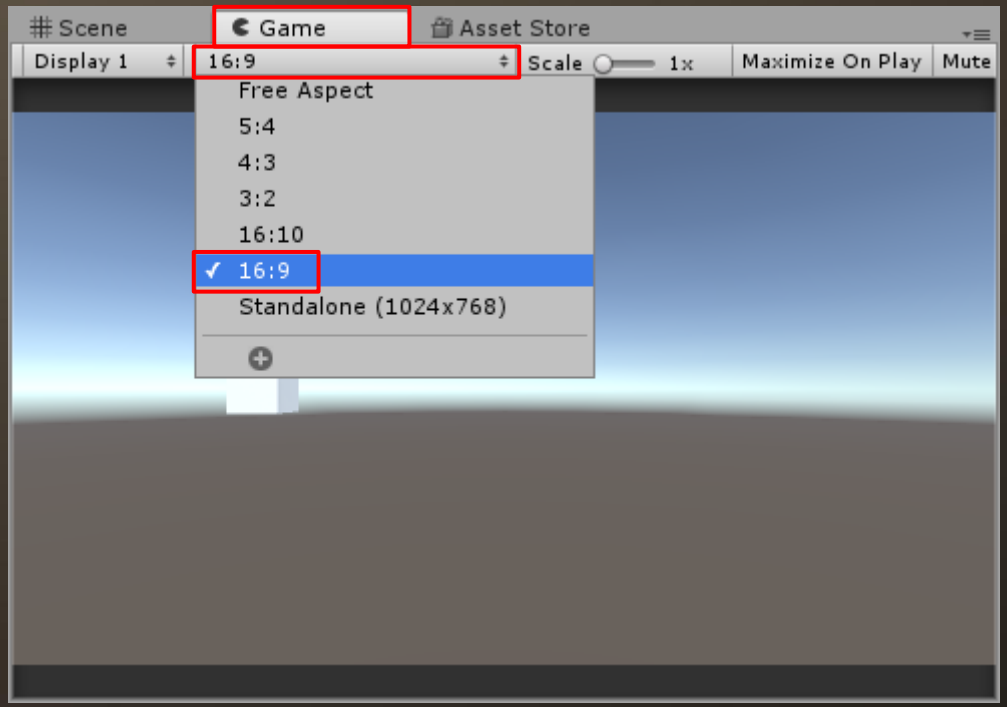
플레이


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

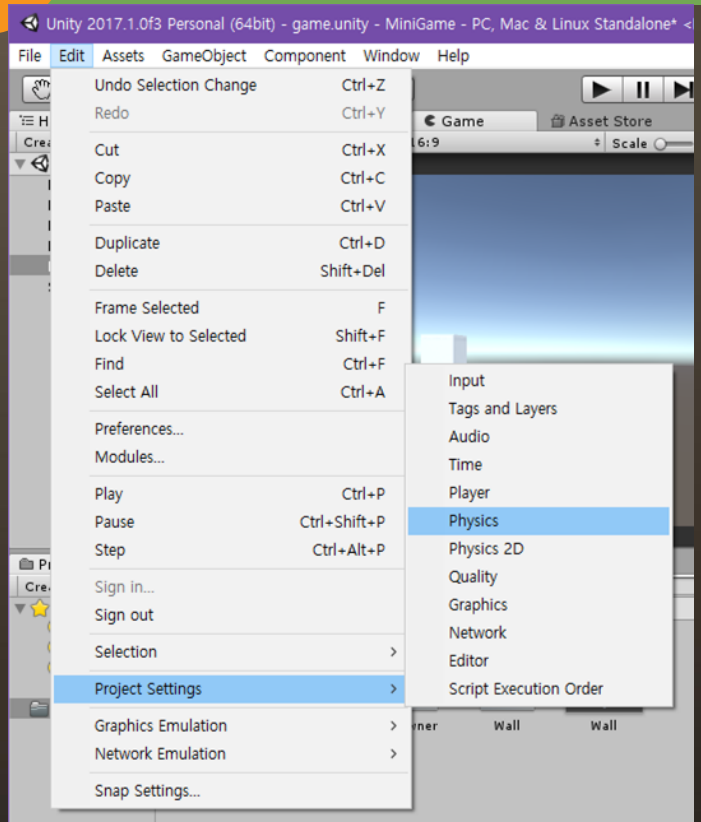
천정 추가하기

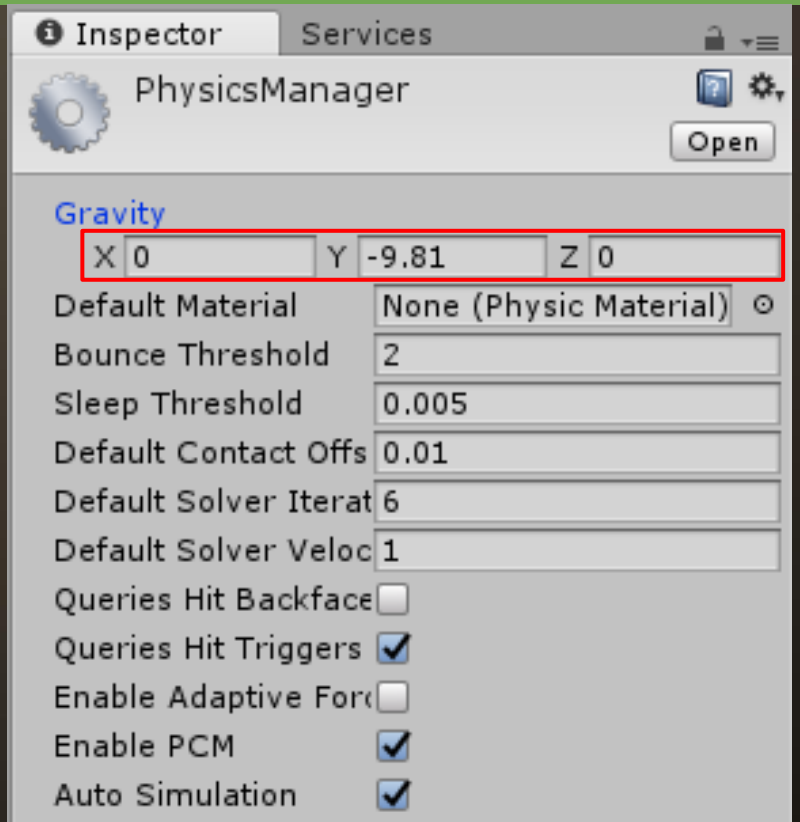

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x
Persp
Inspector
Services
Floor (1)
Static
Tag
Untagged
Layer
Default
Transform
Position X 0 Y 10 Z 0
Rotation X 0 Y 0 Z 0
Scale X 20 Y 1 Z 2
Cube (Mesh Filter)
Box Collider
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*
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...
Input
Tags and Layers
Audio
Time
Player
Physics
Physics 2D
Quality
Graphics
Network
Editor
Script Execution Order
Game
Asset Store
Scale
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

DELICIOUS GAMES

Variations

Try 1
1인칭 시점

시도1: 1인칭 시점

Hierarchy

Create ▾ | All

- game
  - Directional Light
  - Player
    - Main Camera
  - Floor
  - Floor (1)
  - Spawner

Inspector | Services

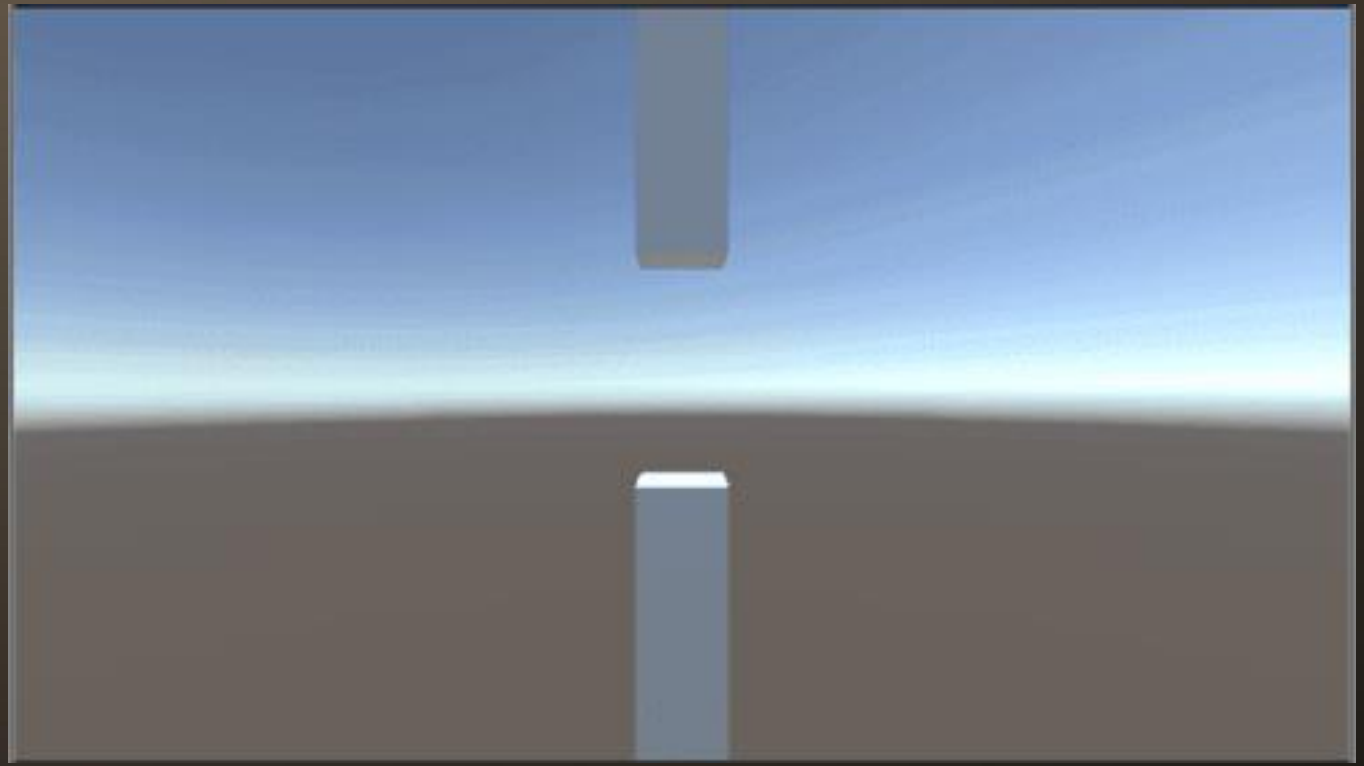
Main Camera ☑ Static

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 () {

    }
```

DELICIOUS GAMES

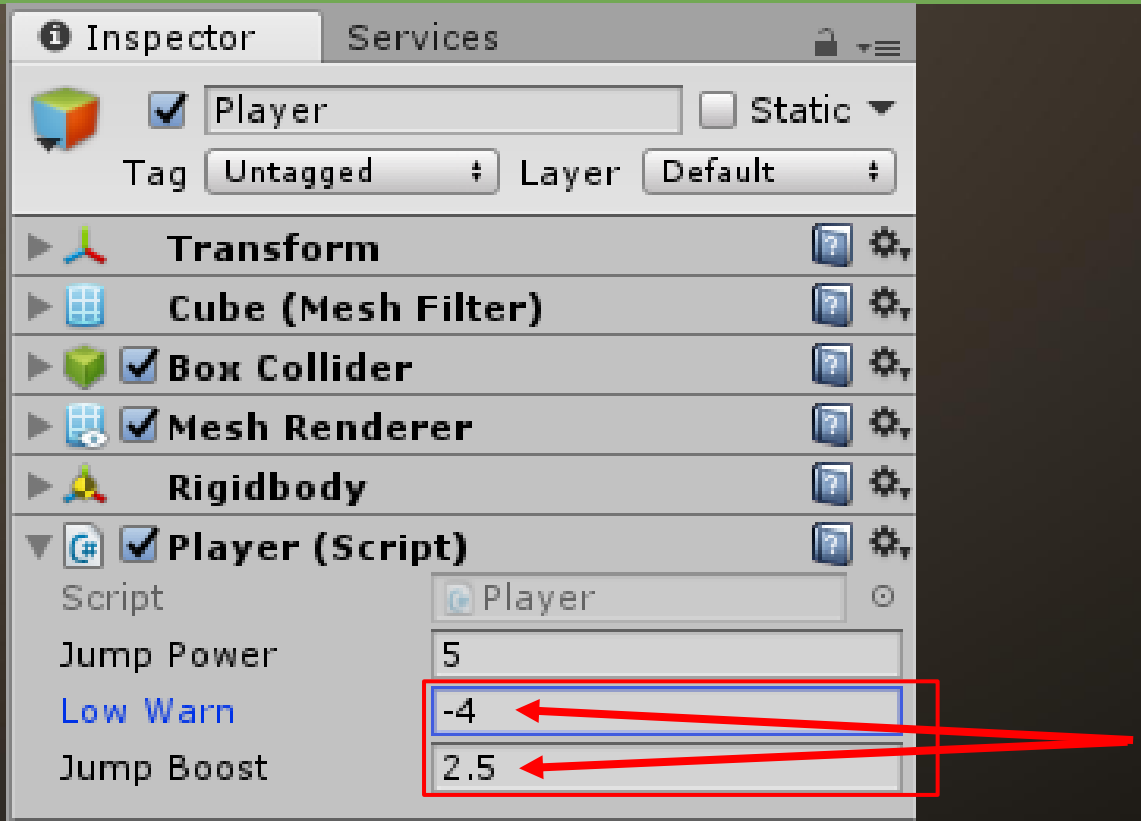
시도2: 낮은 높이에서 점프 부스터

Player.cs

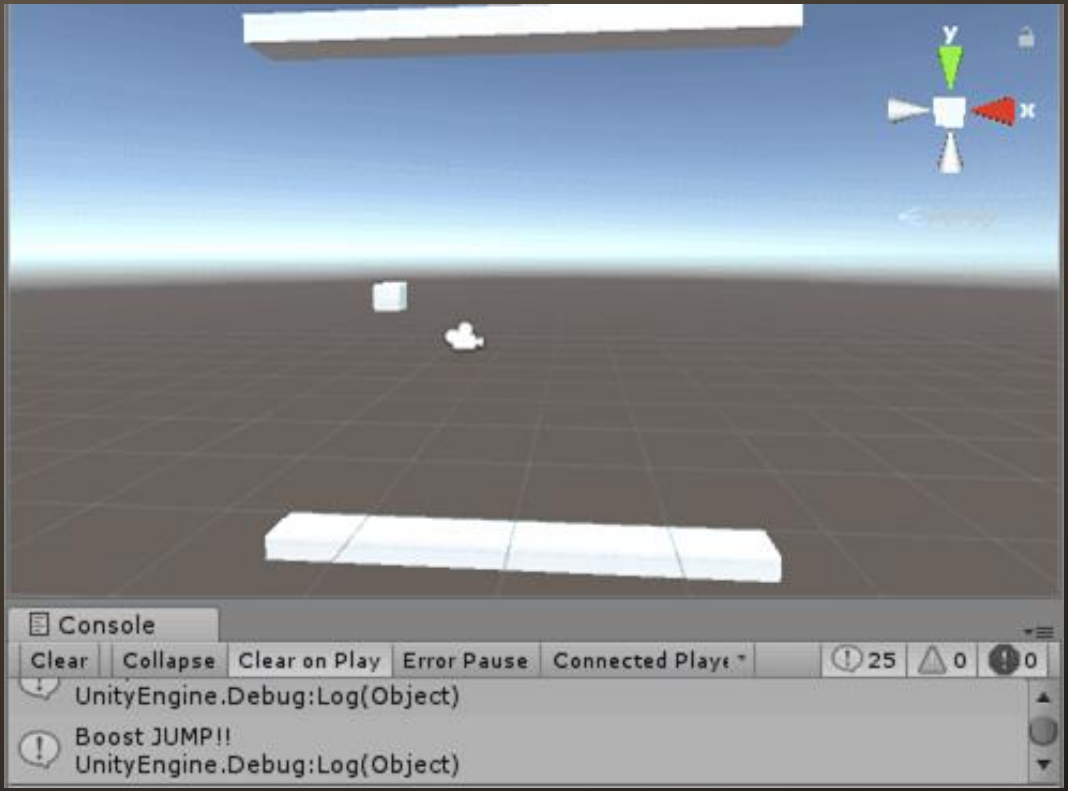
MiniGame | Player | Update()

```
// 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

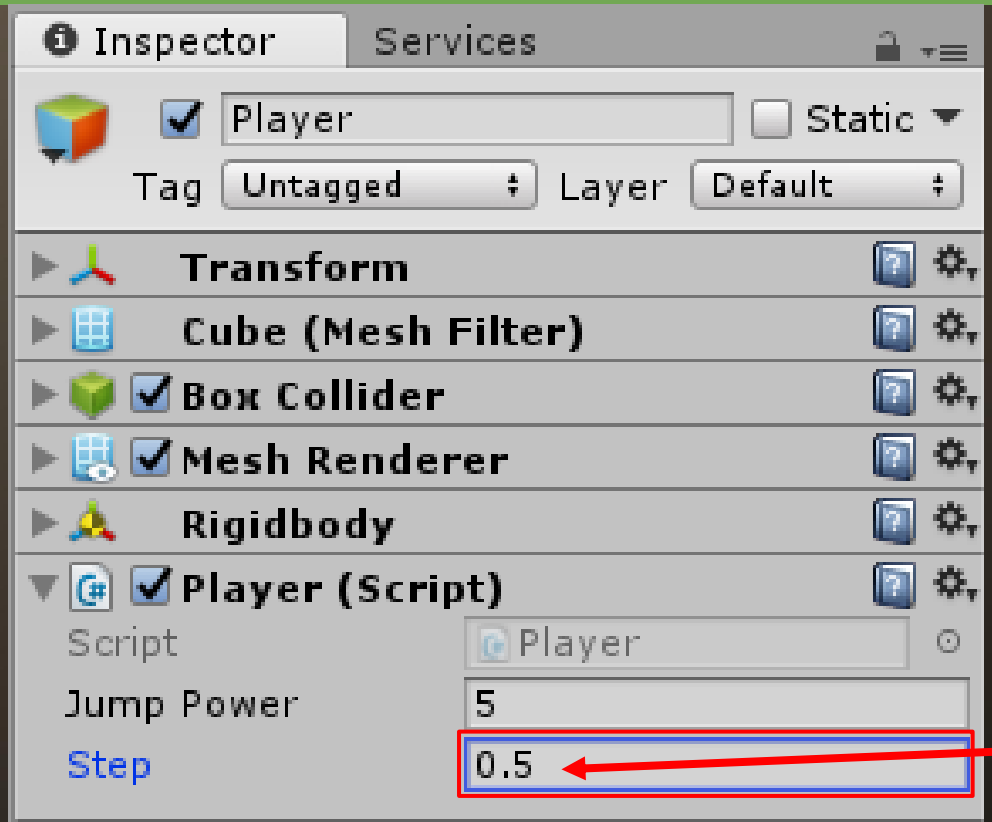
시도3: 점점 앞으로 가는 플레이어

Player.cs

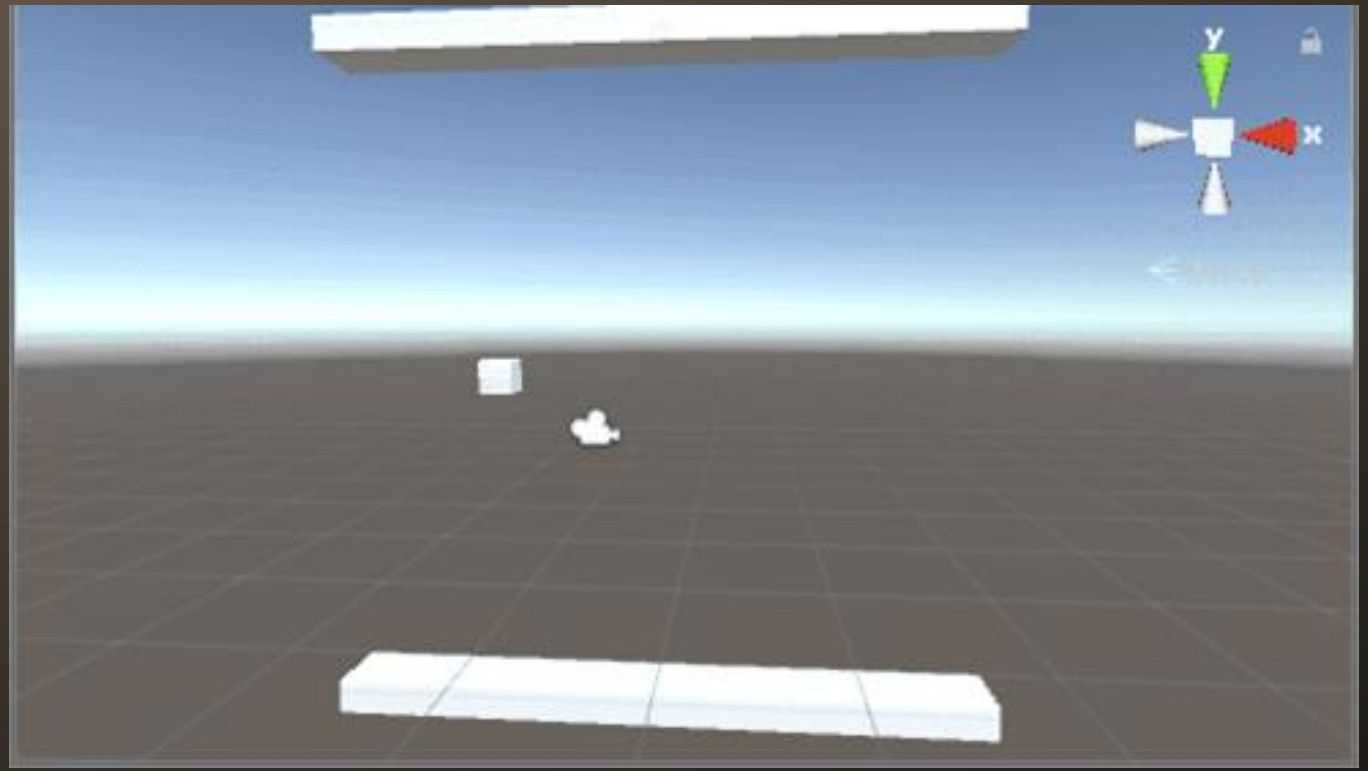
MiniGame | Player | Update()

```
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

DELICIOUS GAMES

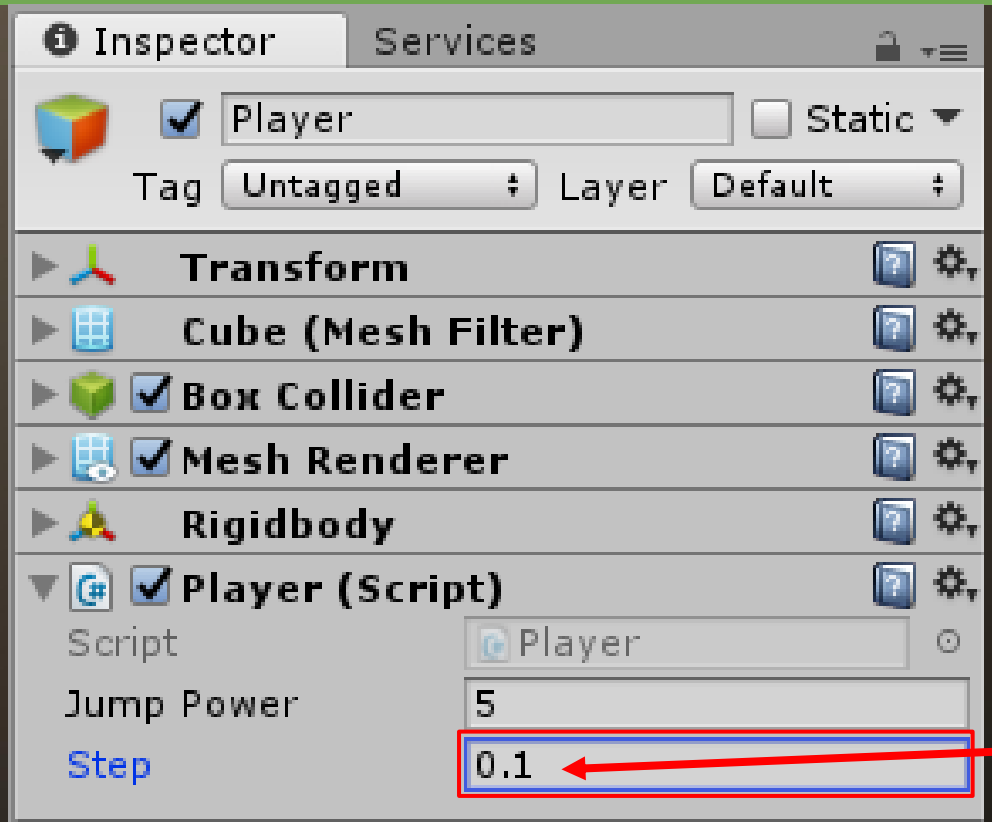
시도4: 점점 키가 크는 플레이어

Player.cs

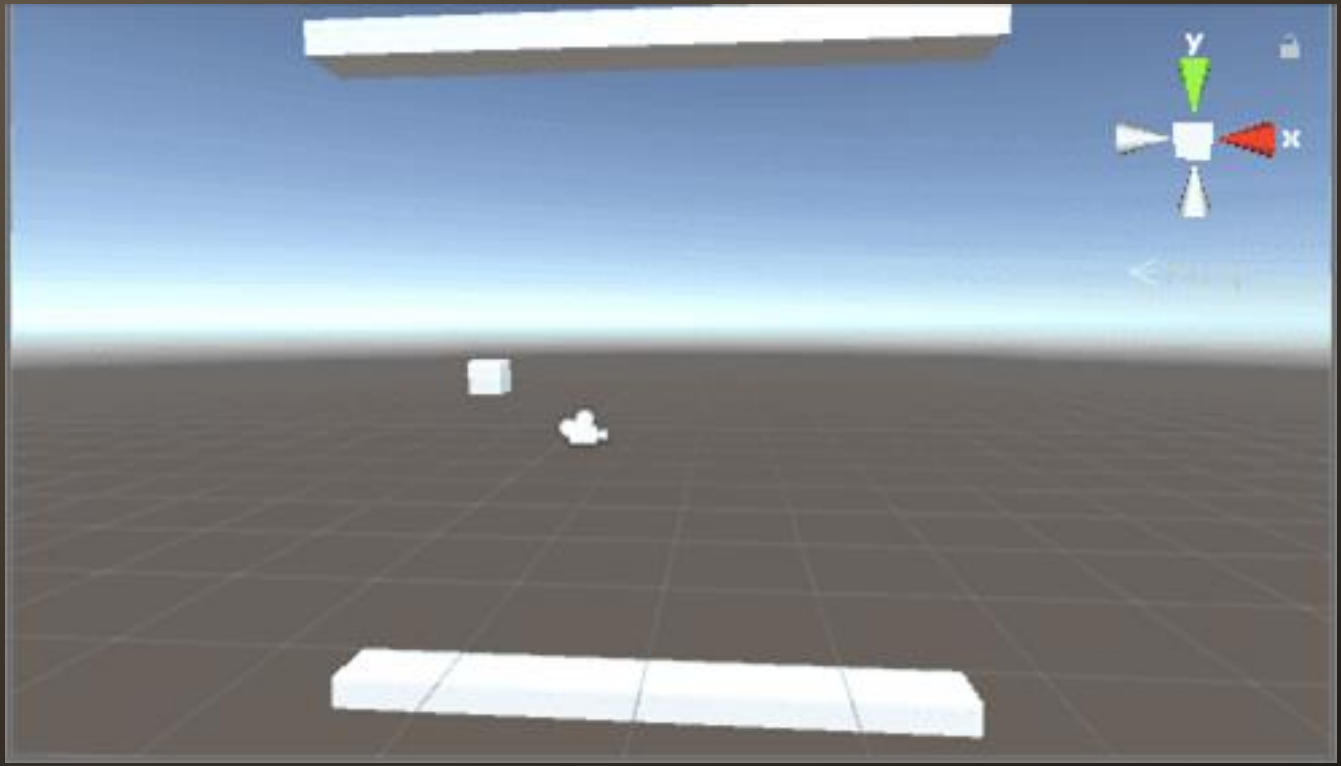
MiniGame | Player | Update()

```
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



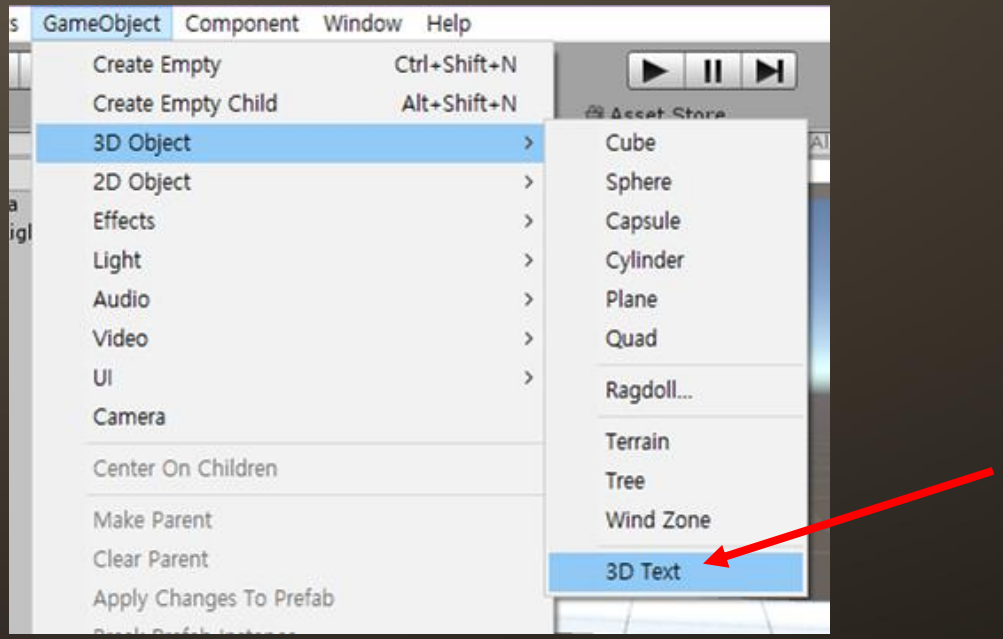
시도4: 점점 키가 크는 플레이어



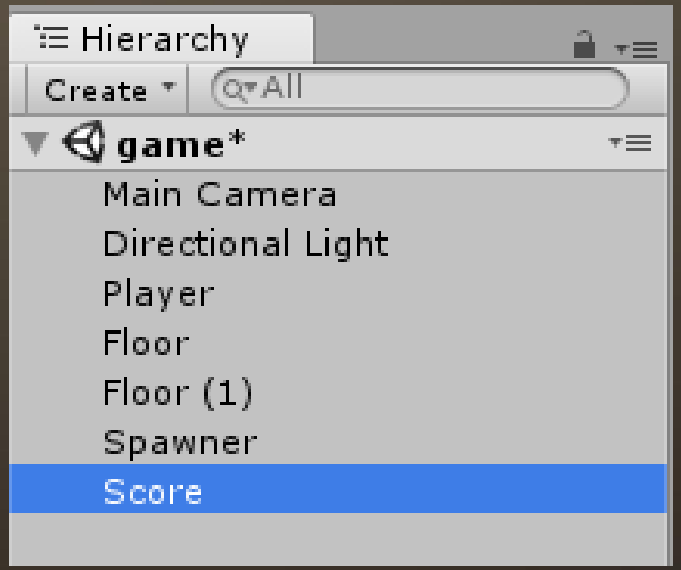
Try 5
점수 카운팅

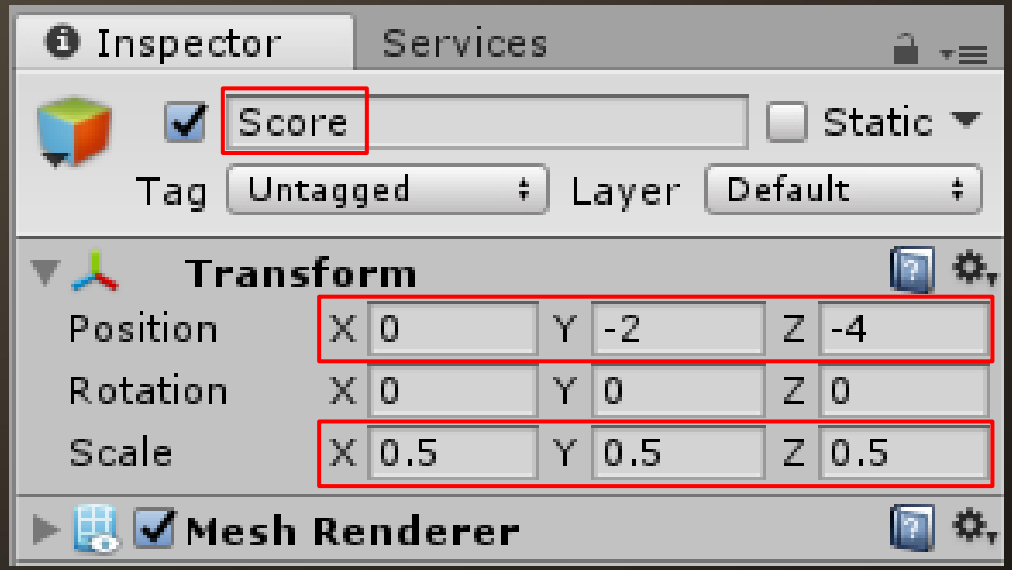
시도5: 점수 카운팅

- GameObject > 3D Object > 3D Text

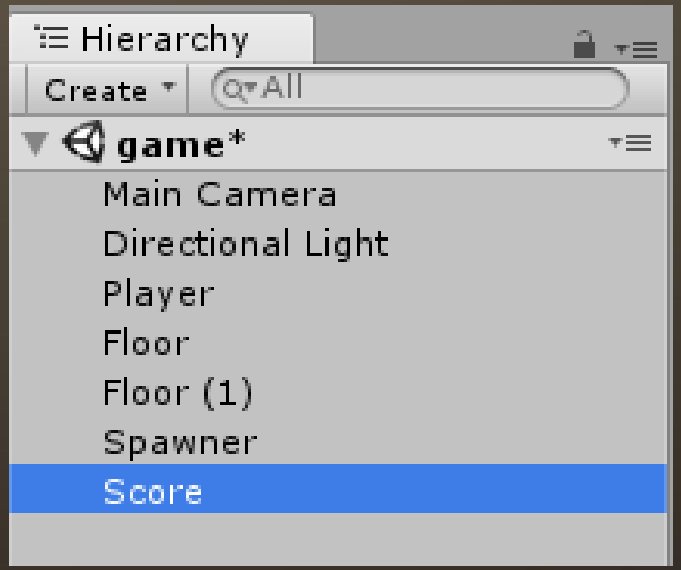


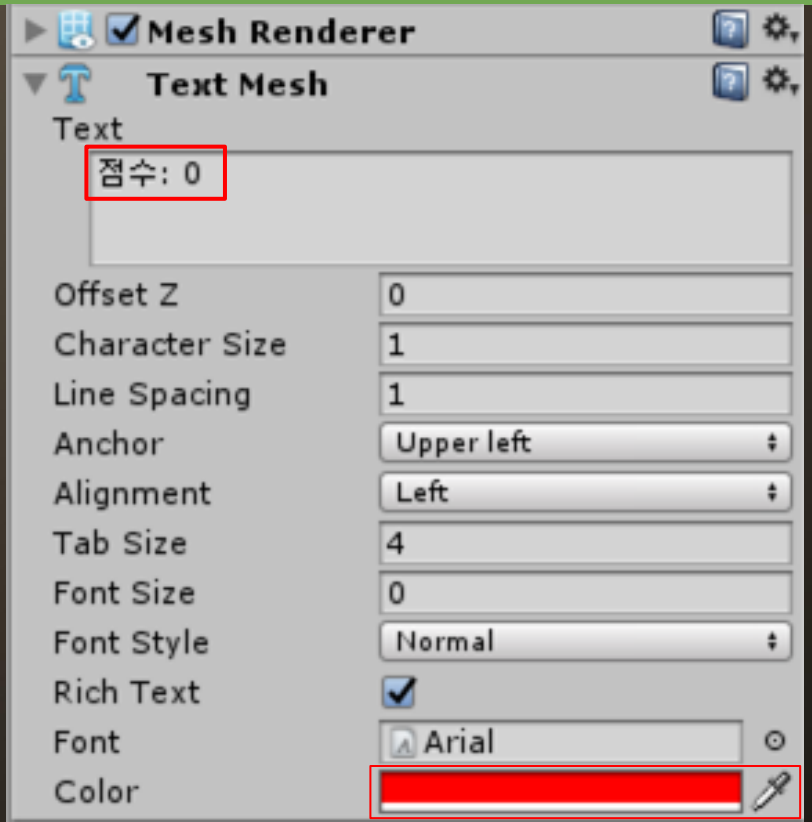
시도5: 점수 카운팅


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 카운팅


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


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
        }

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

시도5: 점수 카운팅

Player.cs

MiniGame | Player | addScore(int s)

```
GetComponent<
Vector3(0, jumpPower, 0);
        }

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

        // 점수 더하기
        public void addScore(int s)
        {
                score += s;
                scoreOutput.text = "점수 : " + score;
        }
}
```

DELICIOUS GAMES

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>();
        }
```

DELICIOUS GAMES

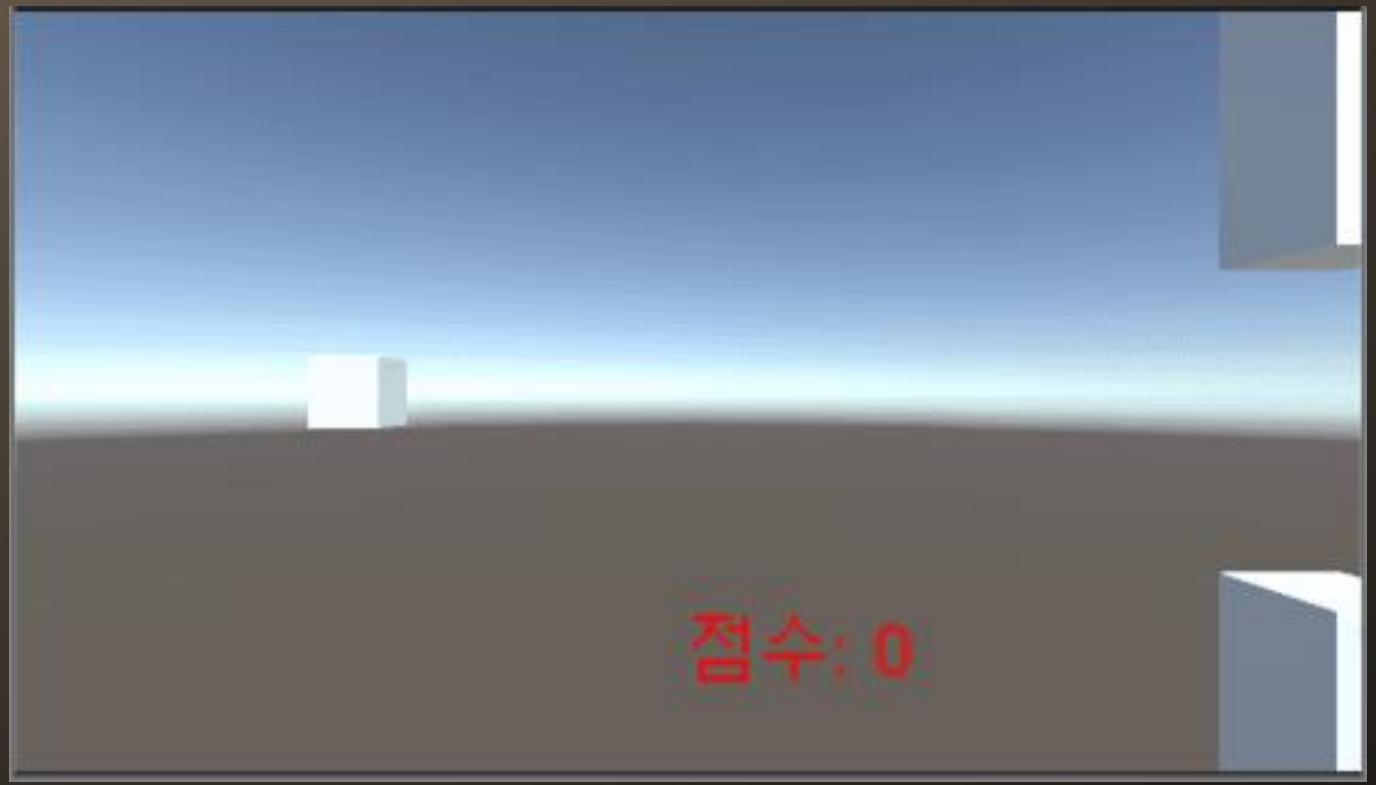
시도5: 점수 카운팅

Wall.cs

MiniGame | Wall | Update()

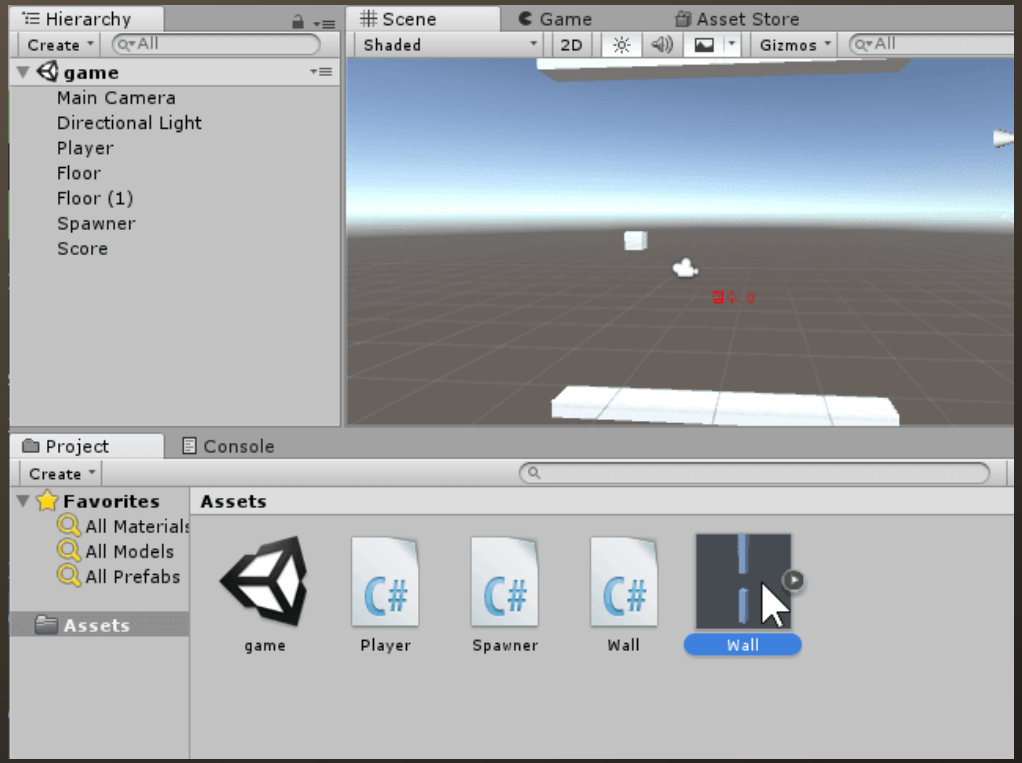
```
        // Update is called once per frame
        void Update () {
            transform.Translate(speed * Time.deltaTime, 0, 0);
            if (transform.position.x < -10)
            {
                Destroy(gameObject);
                player.addScore(1);
            }
        }
}
```

시도5: 점수 카운팅



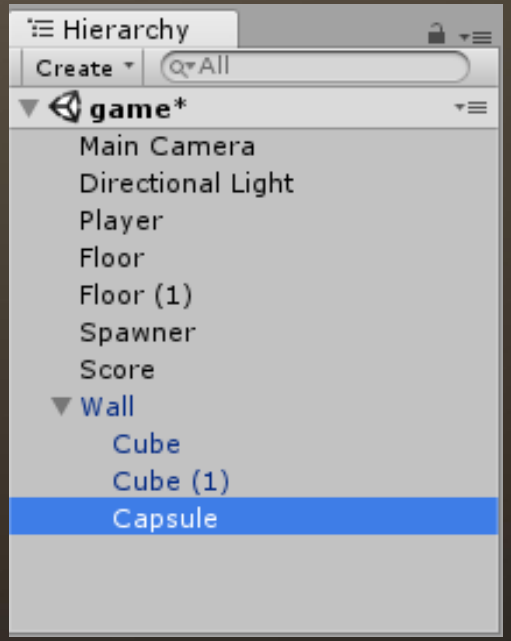
Try 6
다양한 종류의 벽

시도6: 다양한 종류의 벽

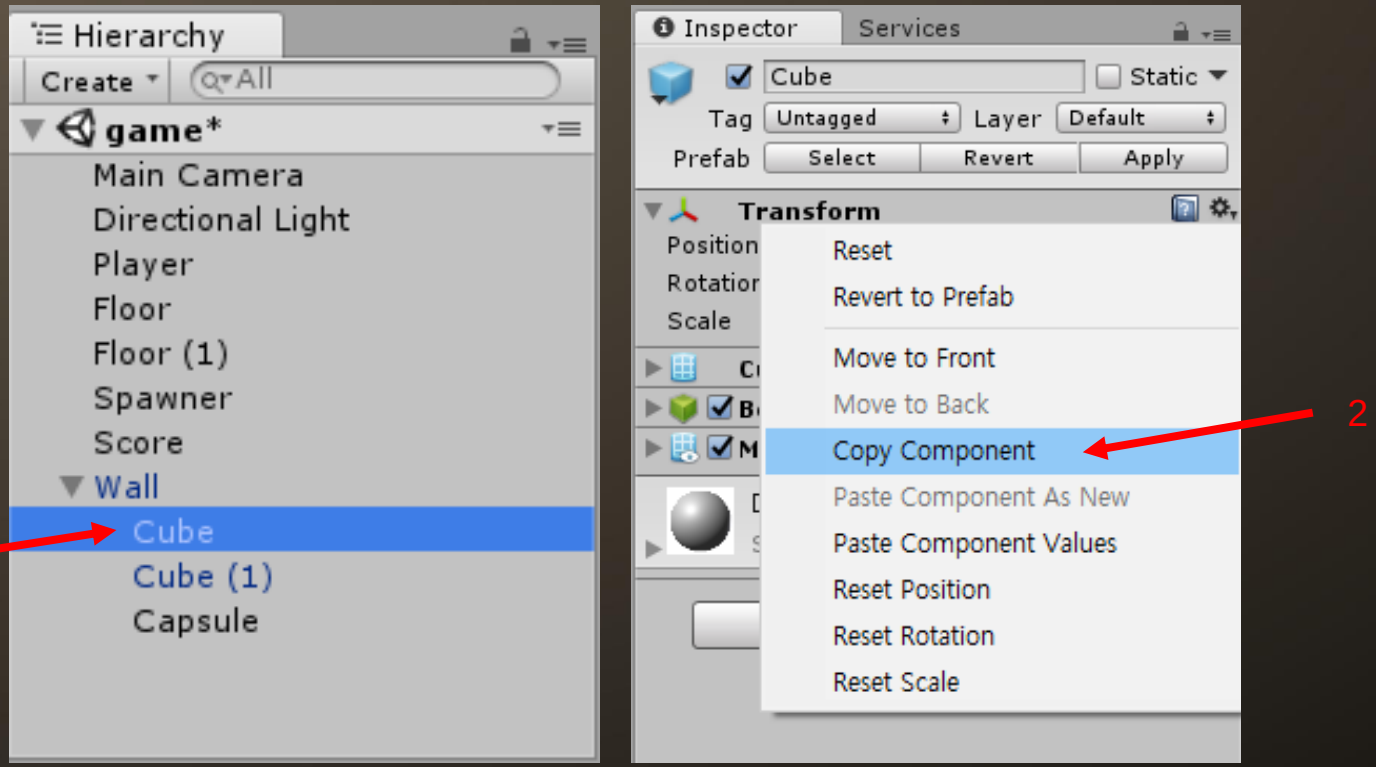

Hierarchy
Create
All
game
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
game
Player
Spawner
Wall
Wall

시도6: 다양한 종류의 벽

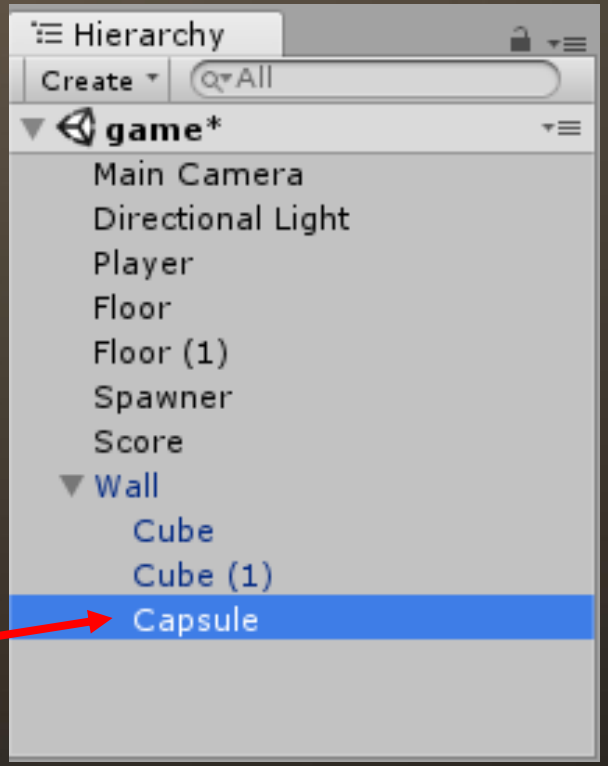
- GameObject > 3D Object > Capsule

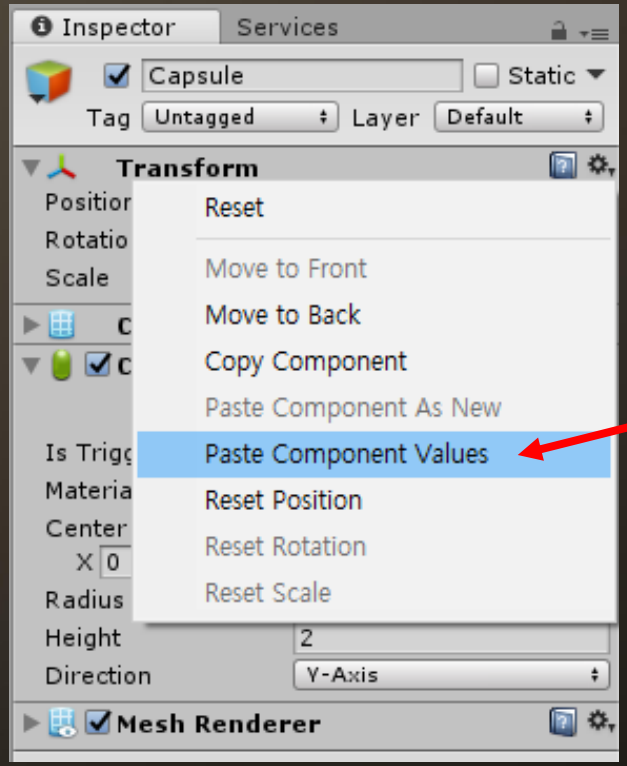


시도6: 다양한 종류의 벽

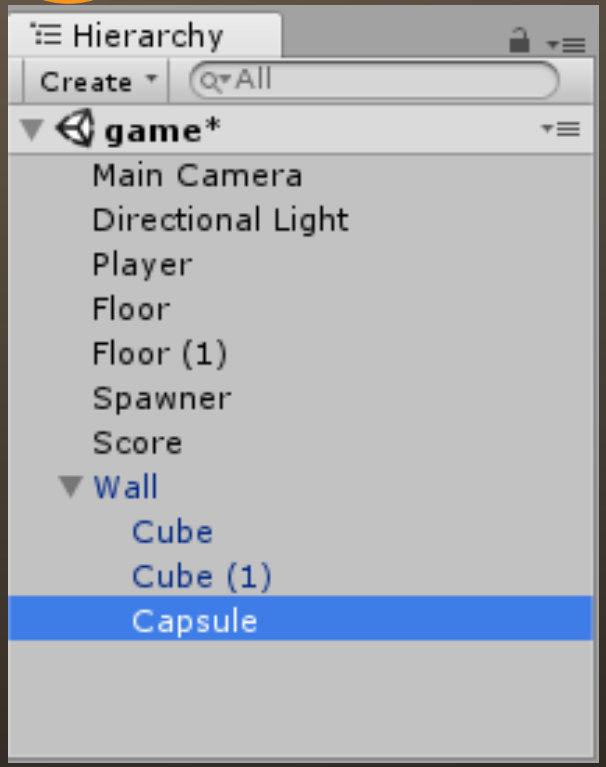

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule
1
Inspector
Services
Cube
Static
Tag
Untagged
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position
Rotation
Scale
Reset
Revert to Prefab
Move to Front
Move to Back
Copy Component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

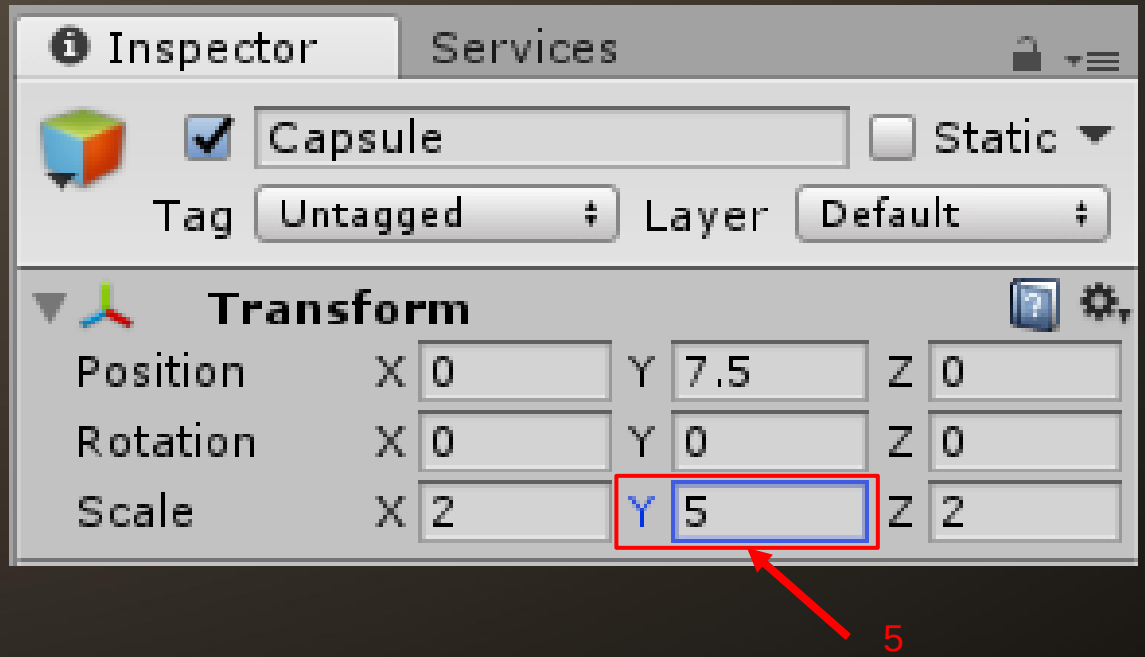
시도6: 다양한 종류의 벽


Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule
3

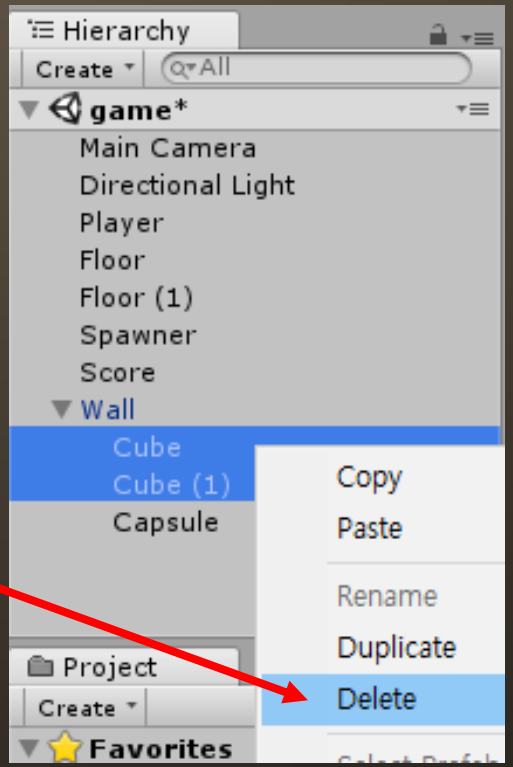

Inspector
Services
Capsule
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Reset
Move to Front
Move to Back
Copy Component
Paste Component As New
Paste Component Values
Reset Position
Reset Rotation
Reset Scale
Is Trigg
Material
Center
X 0
Radius
Height
2
Direction
Y-Axis
Mesh Renderer
4

시도6: 다양한 종류의 벽

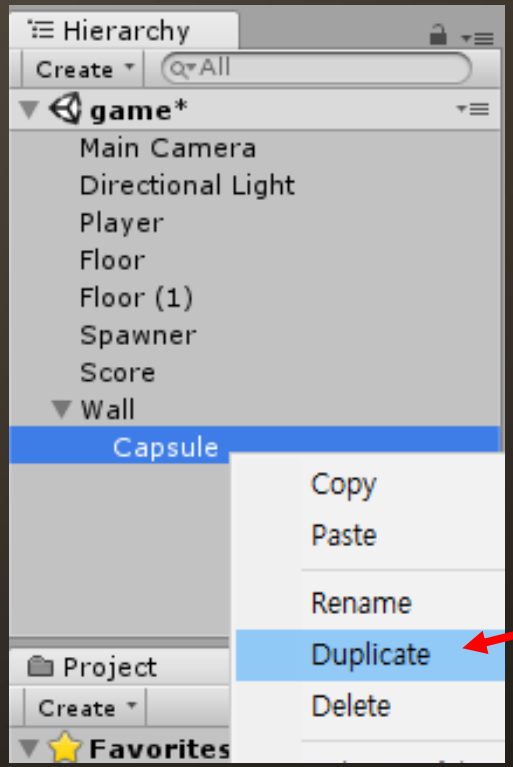

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Cube
Cube (1)
Capsule


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

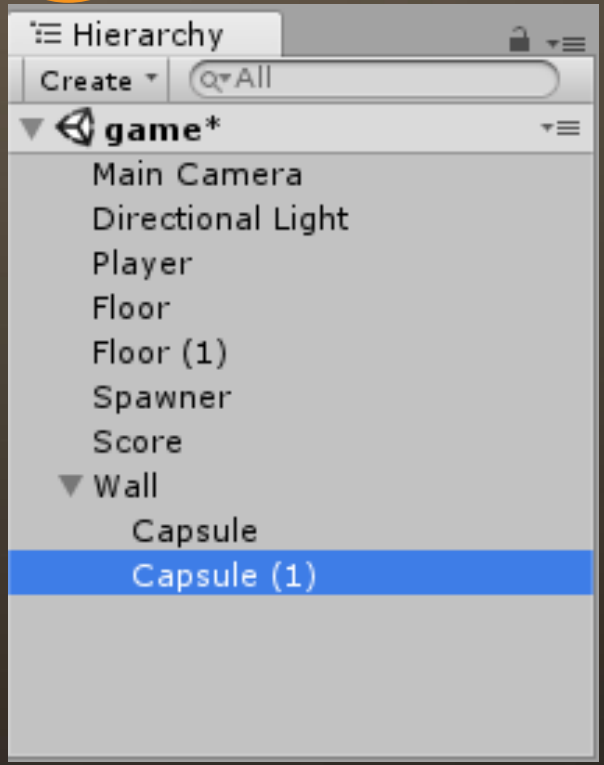


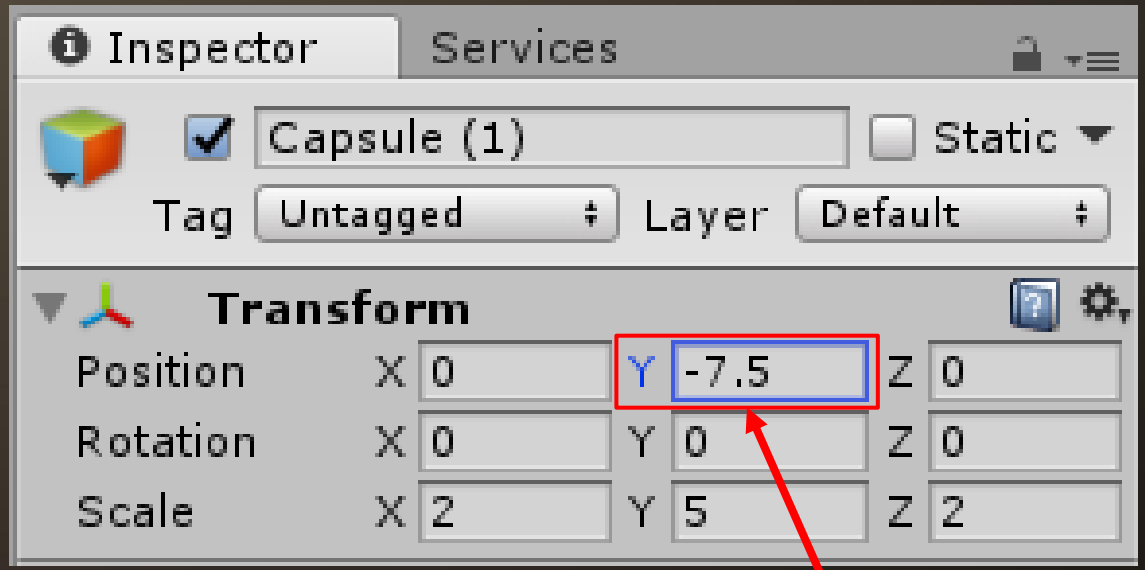
6



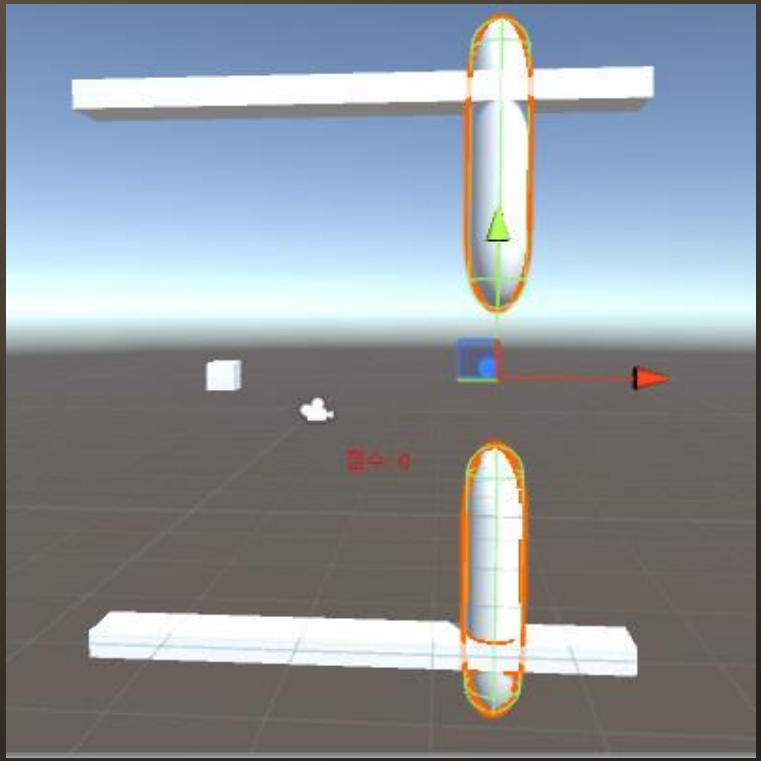
7

시도6: 다양한 종류의 벽

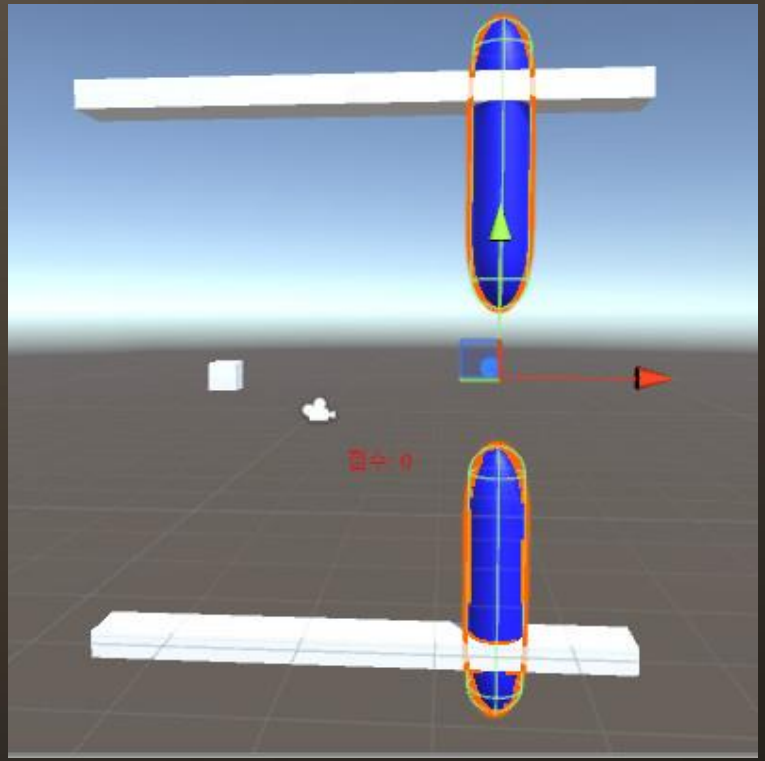

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall
Capsule
Capsule (1)


Inspector
Services
Capsule (1)
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

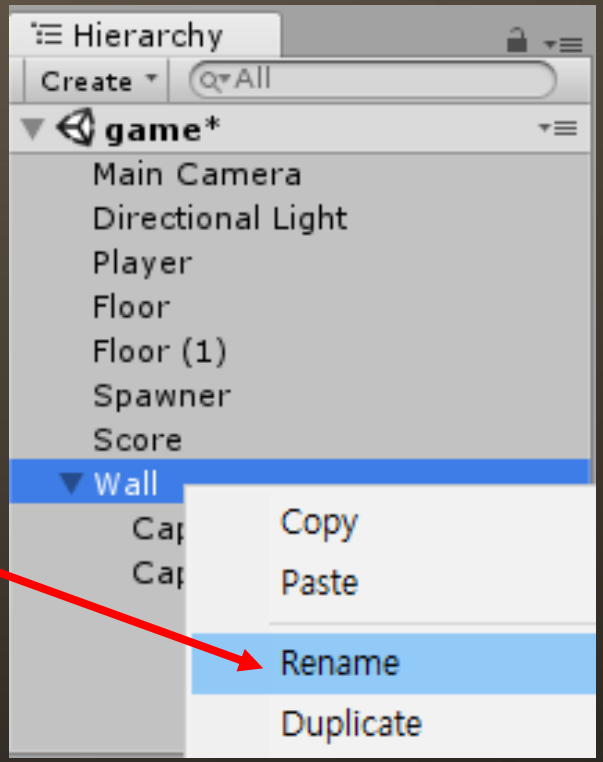
시도6: 다양한 종류의 벽



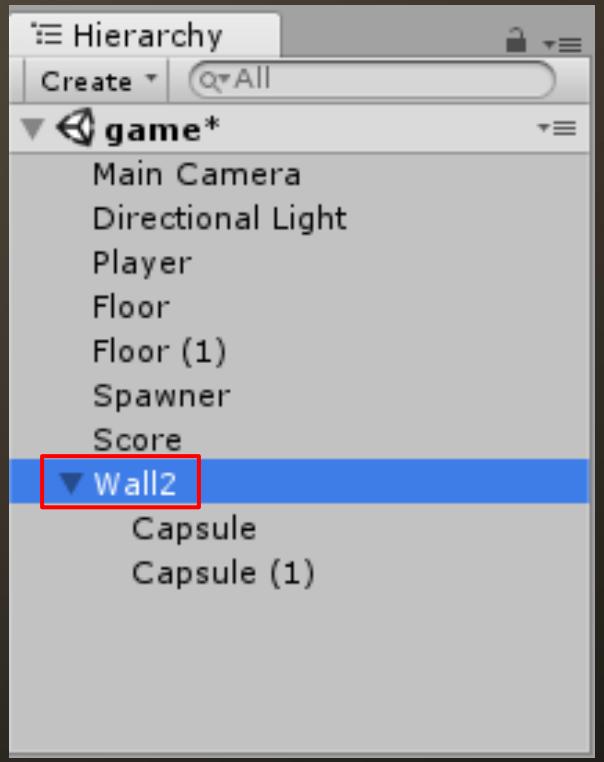
시도6: 다양한 종류의 벽



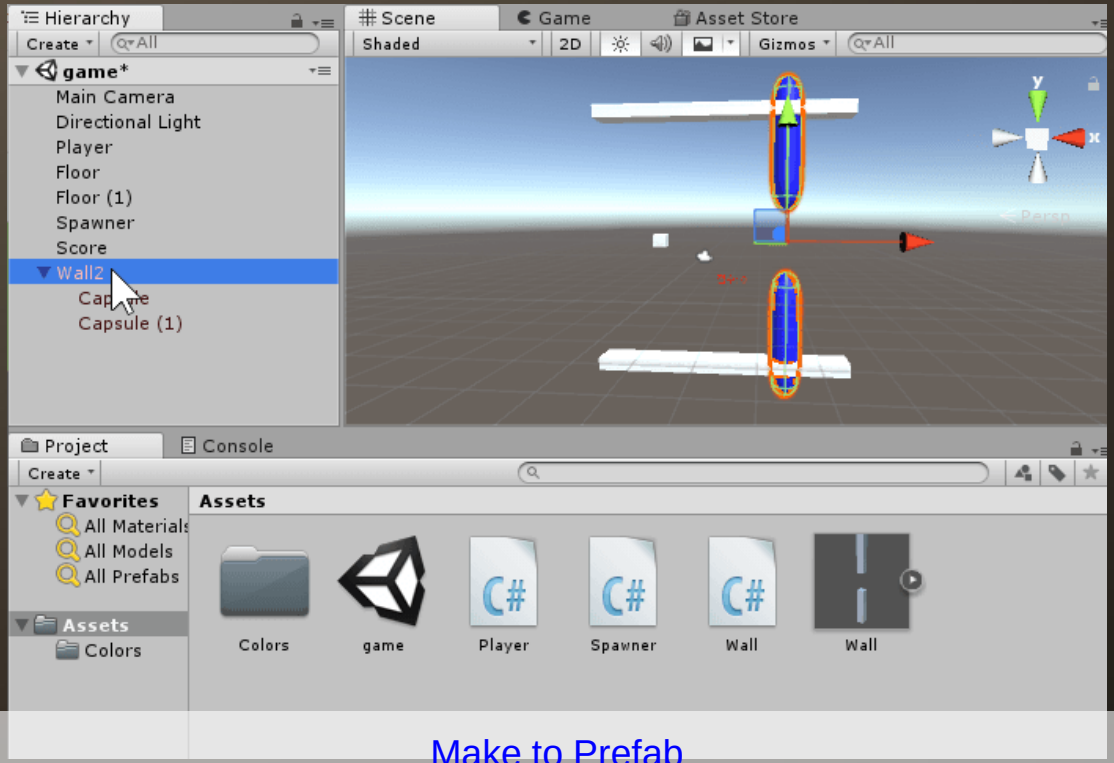
시도6: 다양한 종류의 벽



9

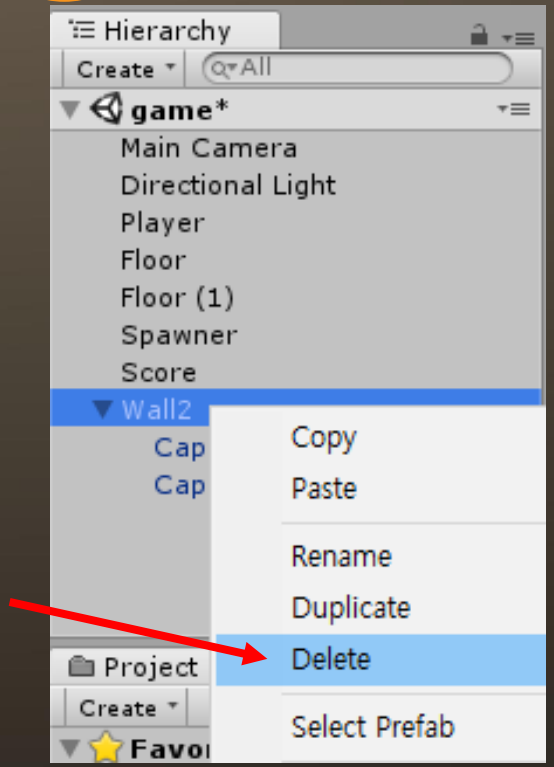


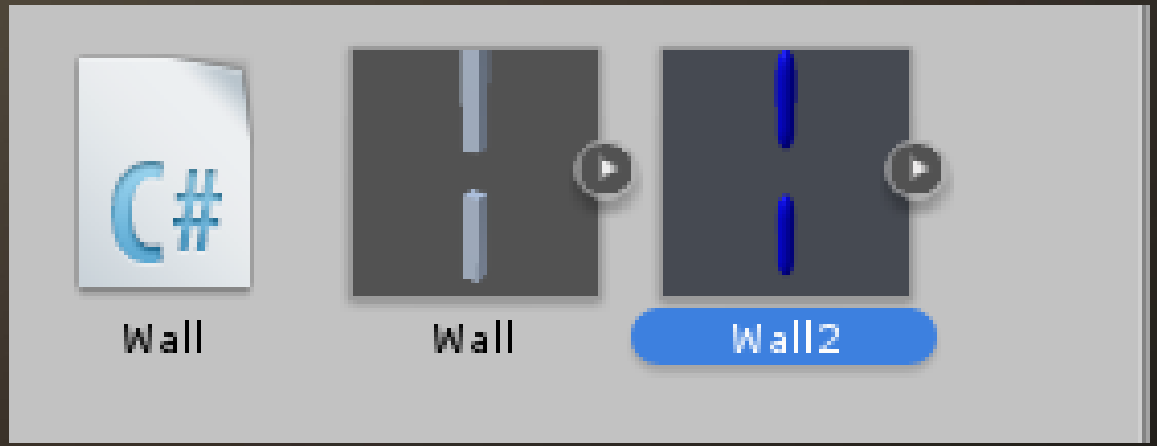
시도6: 다양한 종류의 벽



Make to Prefab

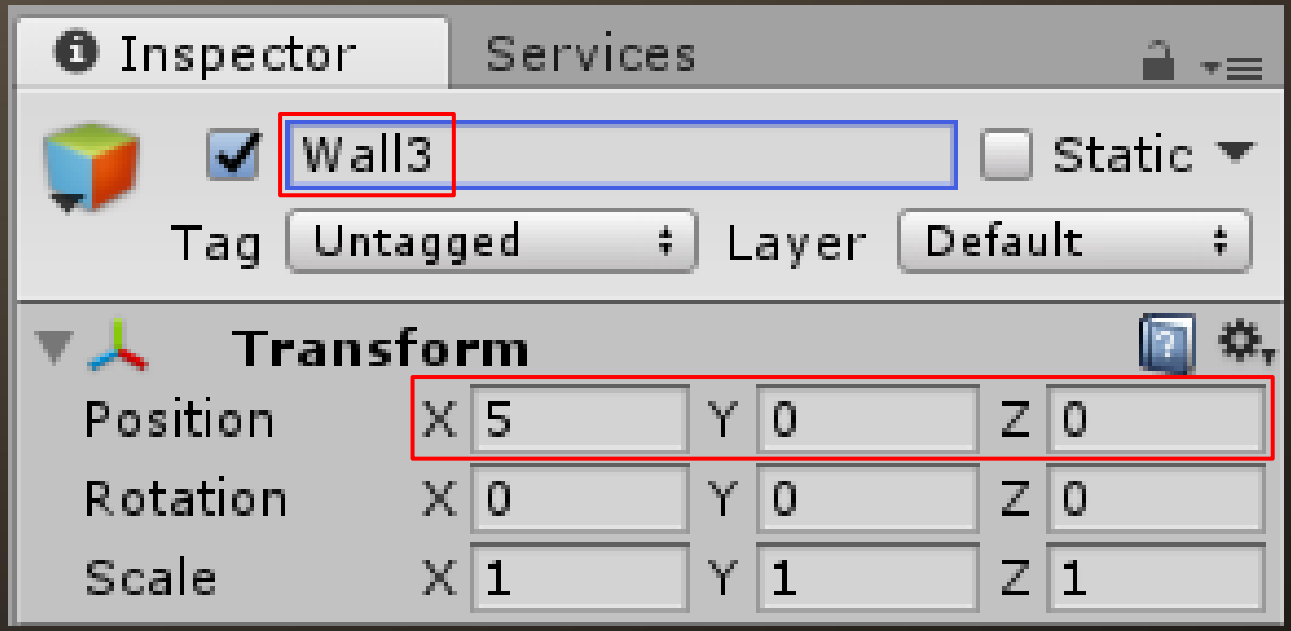
시도6: 다양한 종류의 벽


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

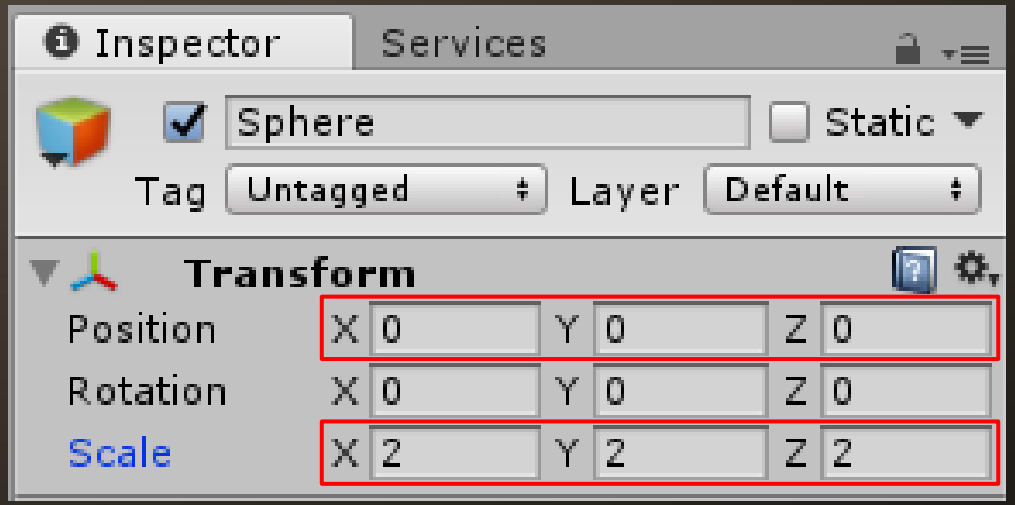
시도6: 다양한 종류의 벽

- GameObject > Create Empty



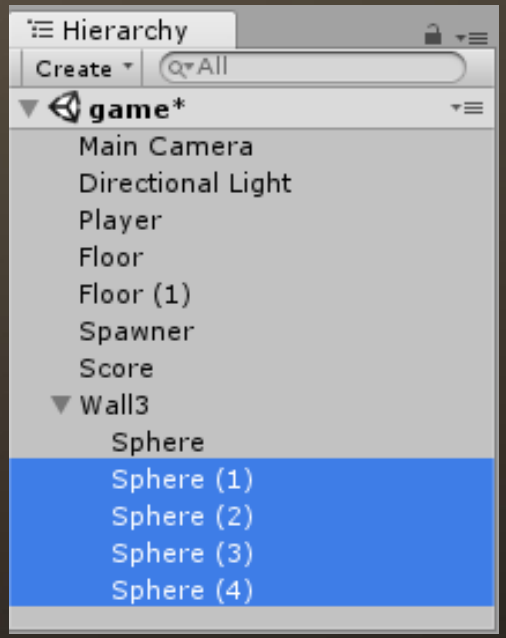
시도6: 다양한 종류의 벽

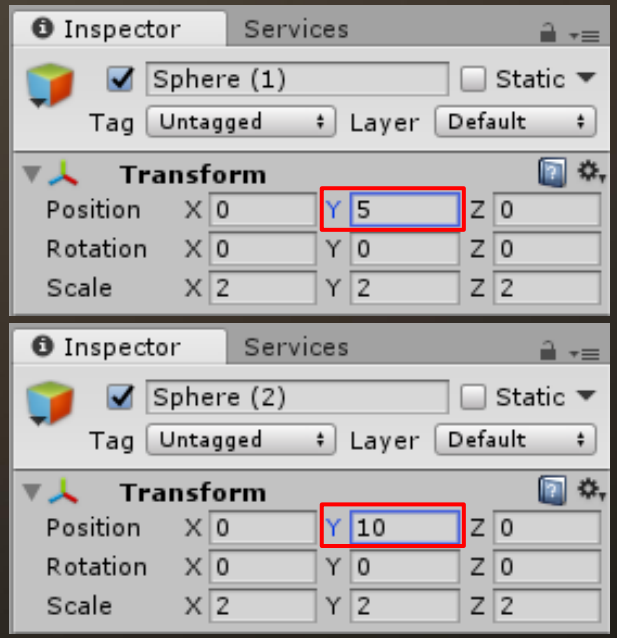
- GameObject > 3D Object > Sphere



시도6: 다양한 종류의 벽

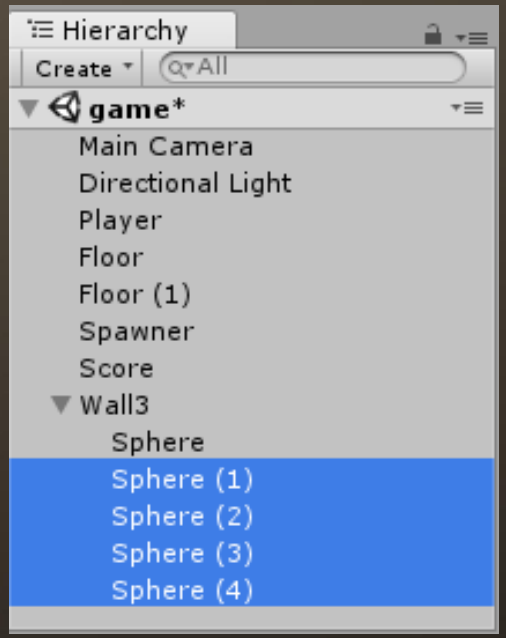
- Sphere > Duplicate

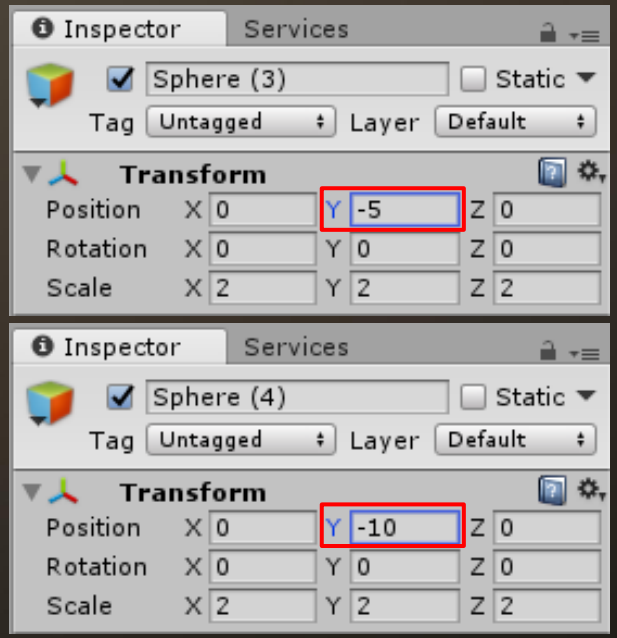




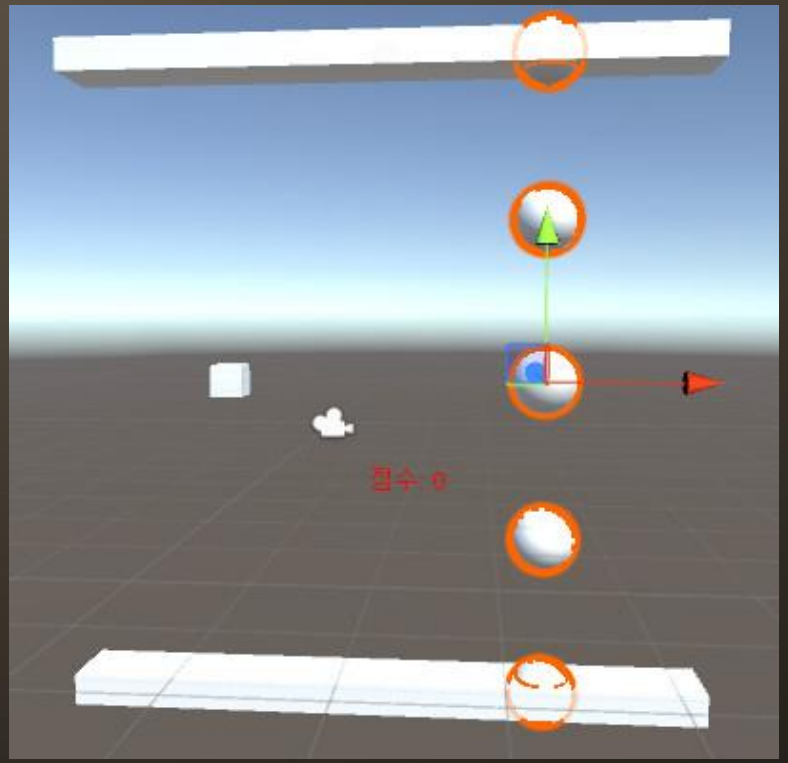
시도6: 다양한 종류의 벽

- Sphere > Duplicate

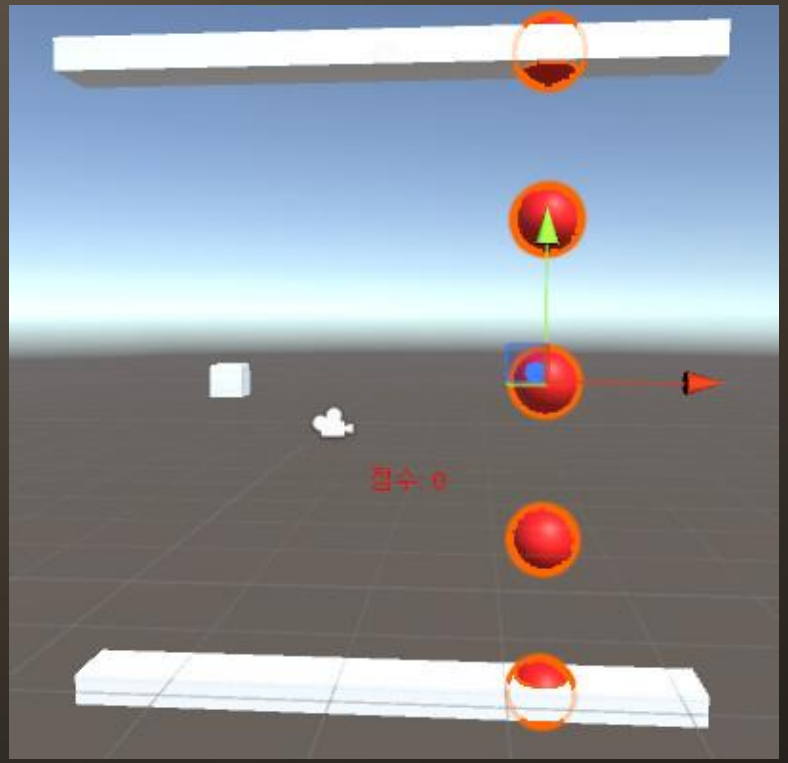




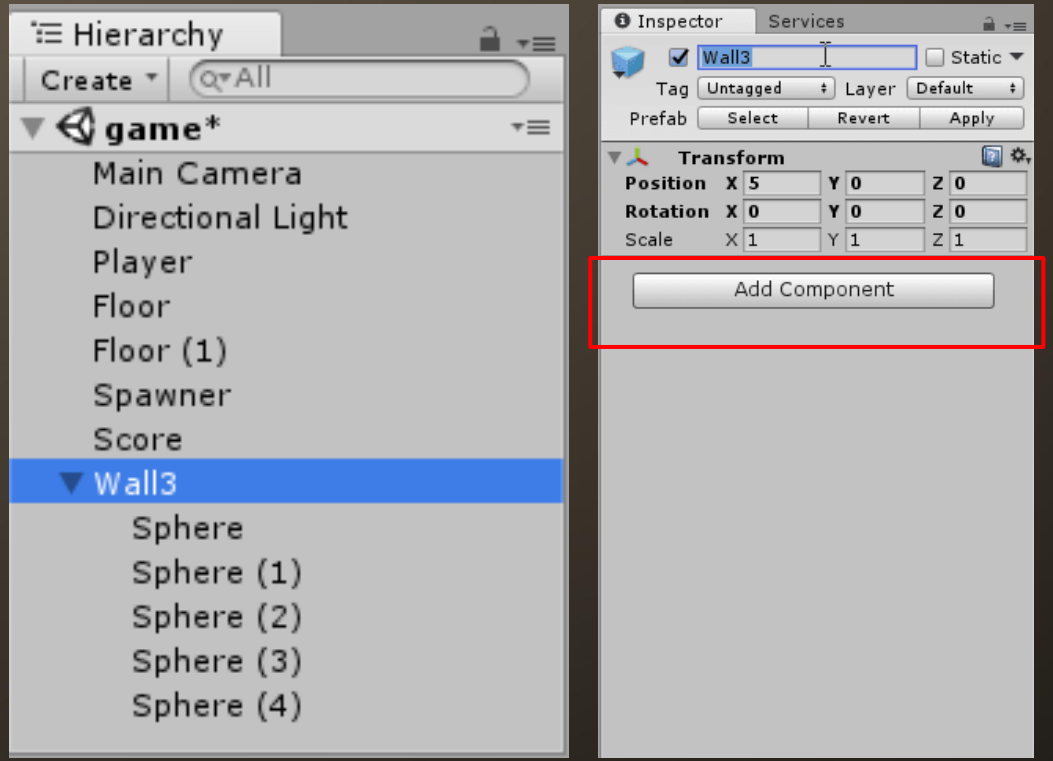
시도6: 다양한 종류의 벽



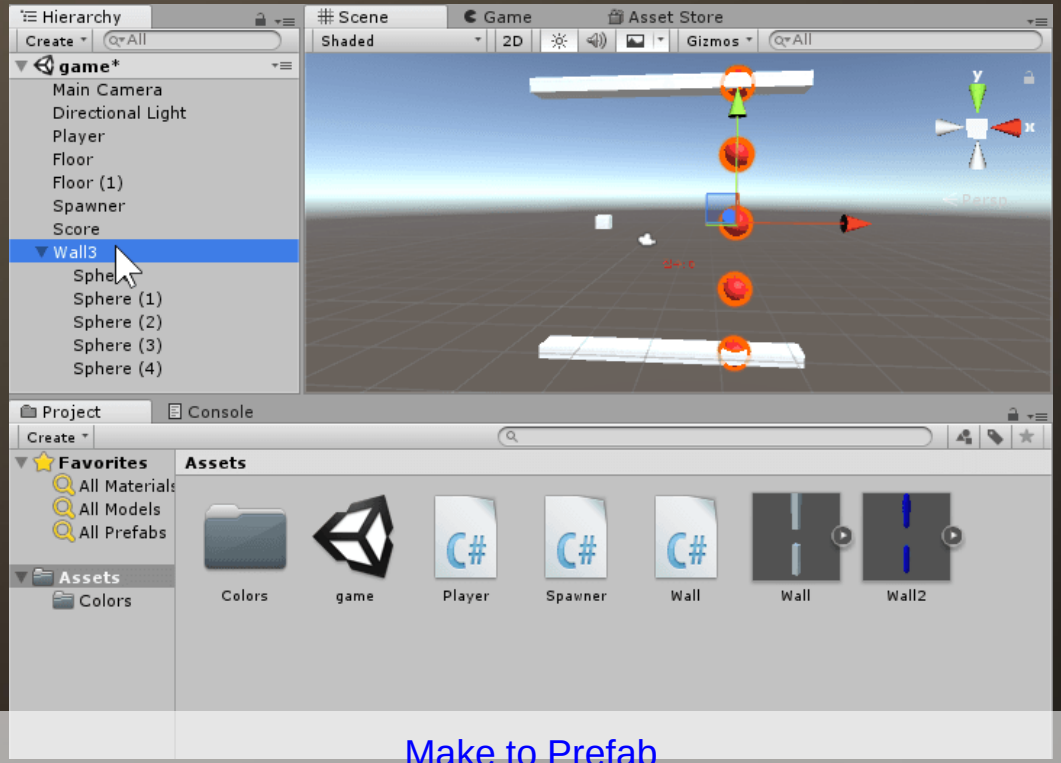
시도6: 다양한 종류의 벽



시도6: 다양한 종류의 벽

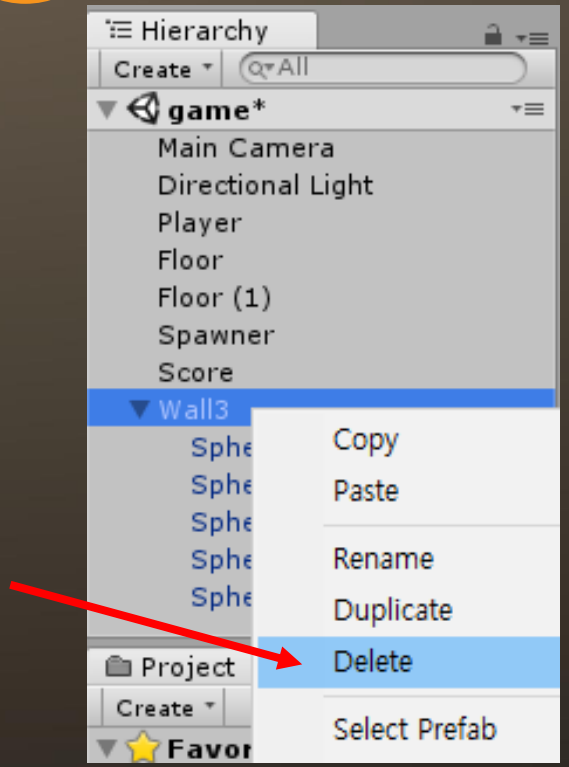

Hierarchy
Create
All
game*
Main Camera
Directional Light
Player
Floor
Floor (1)
Spawner
Score
Wall3
Sphere
Sphere (1)
Sphere (2)
Sphere (3)
Sphere (4)
Inspector
Services
Wall3
Static
Tag
Untagged
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position X 5 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Add Component

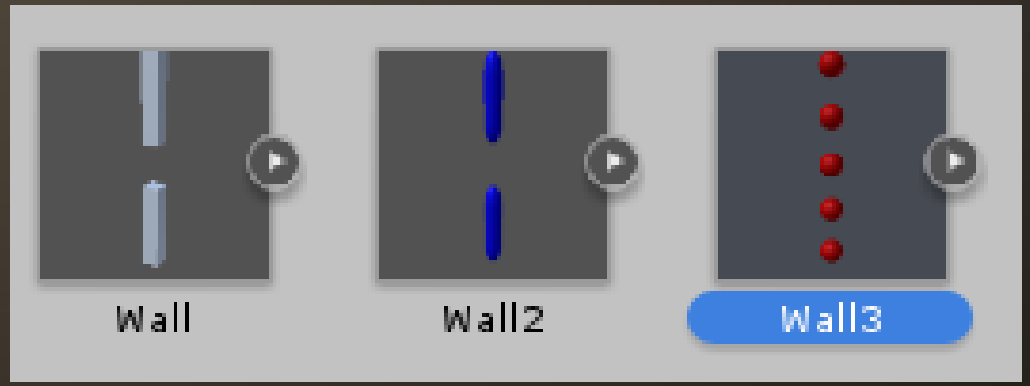
시도6: 다양한 종류의 벽



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

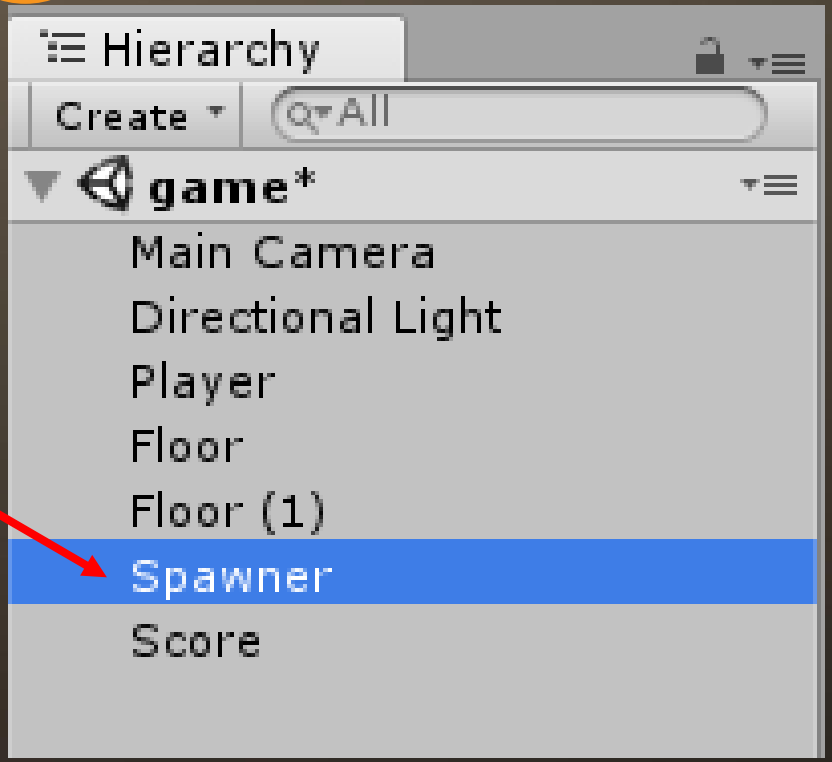
시도6: 다양한 종류의 벽

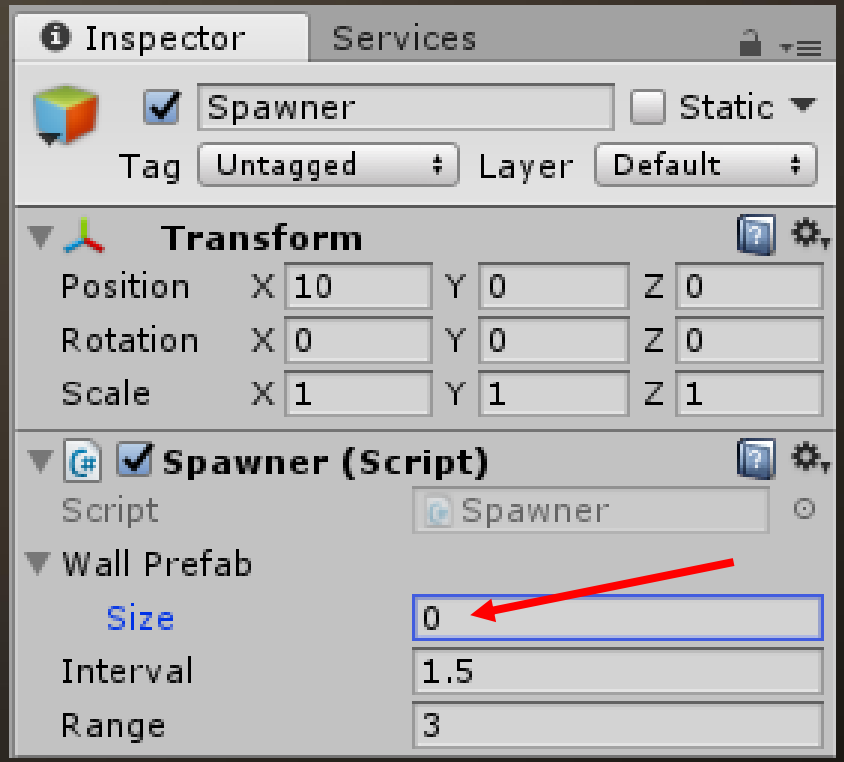
```
public class Spawner : MonoBehaviour {

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

시도6: 다양한 종류의 벽

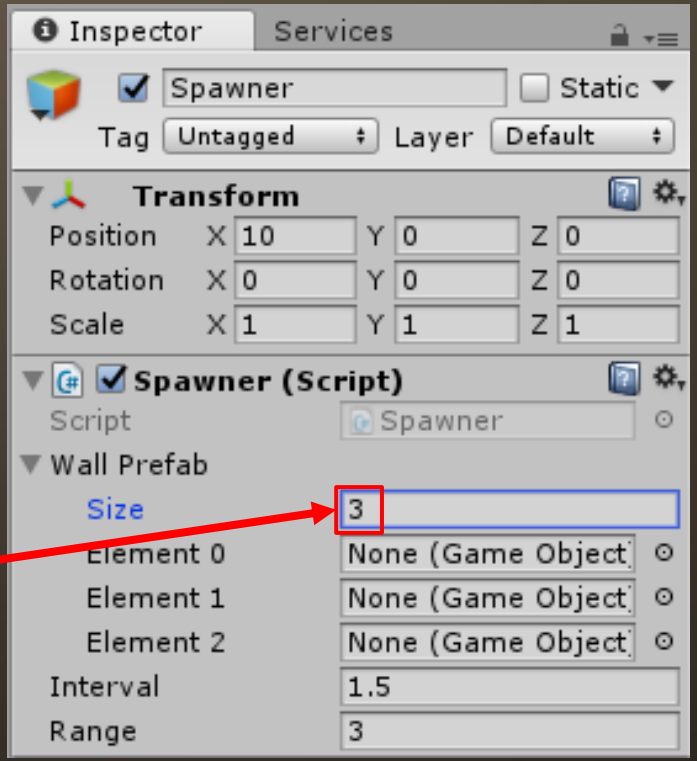
```
        // Update is called once per frame
        void Update () {
                term += Time.deltaTime;
                if (term >= interval)
                {
                        Vector3 pos = transform.position;
                        pos.y += Random.Range(-range, range);
            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

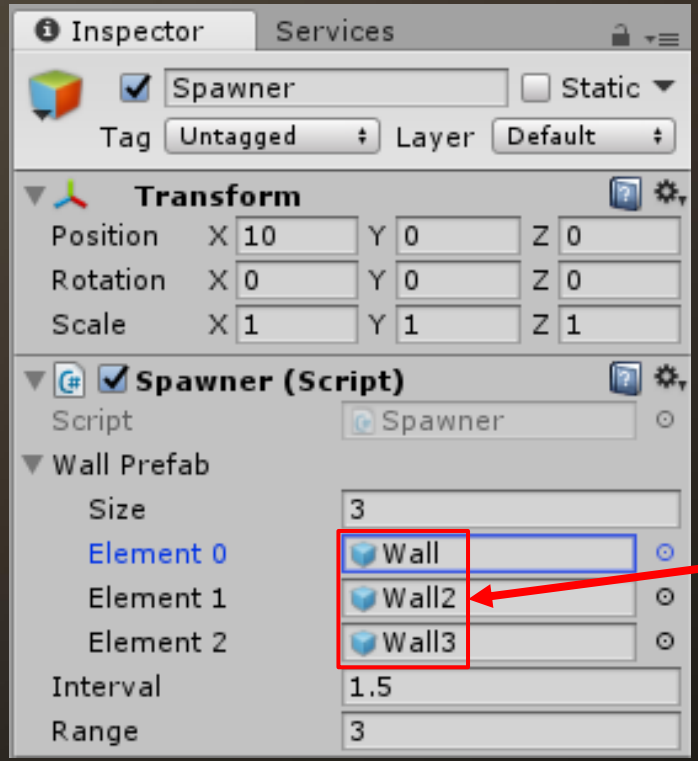
시도6: 다양한 종류의 벽


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

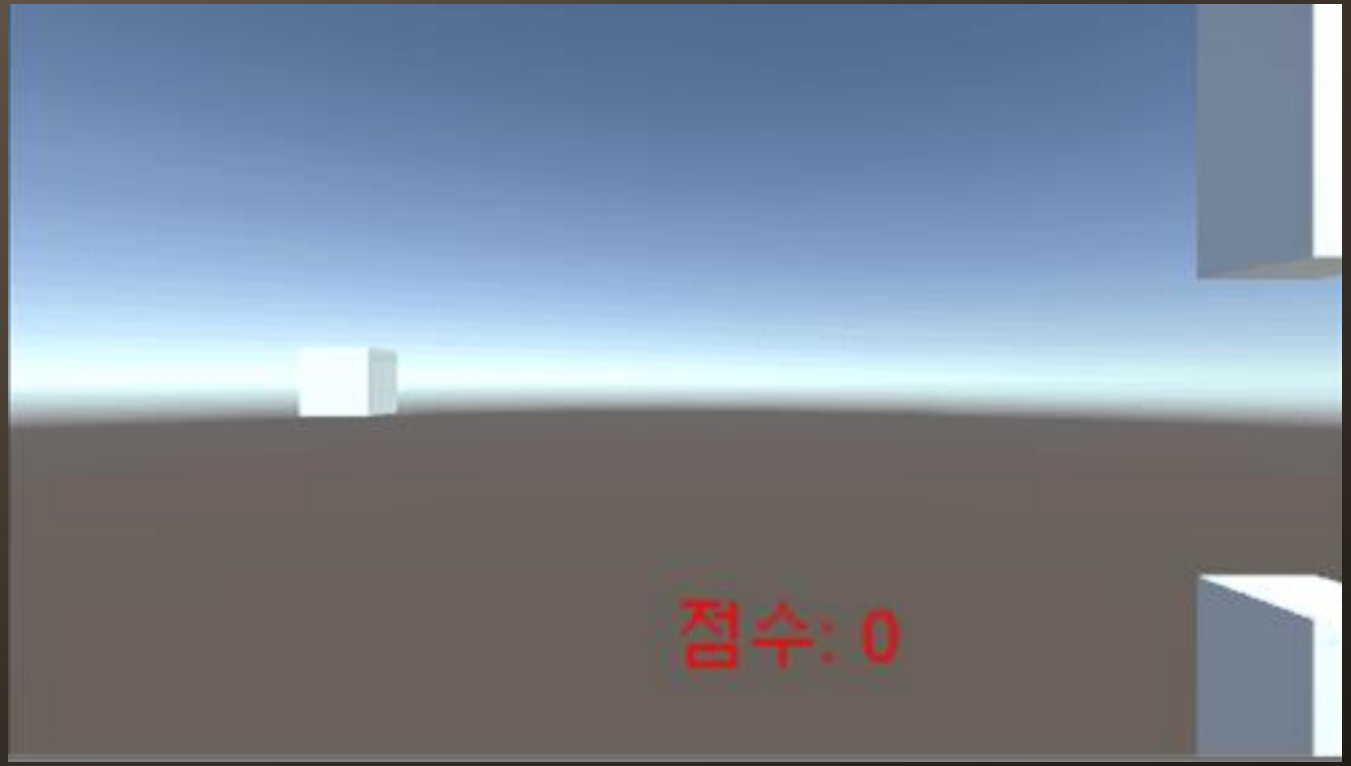

Inspector
Services
Spawner
Static
Tag
Untagged
Layer
Default
Transform
Position X 10 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Spawner (Script)
Script
Spawner
Wall Prefab
Size
0
Interval
1.5
Range
3

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

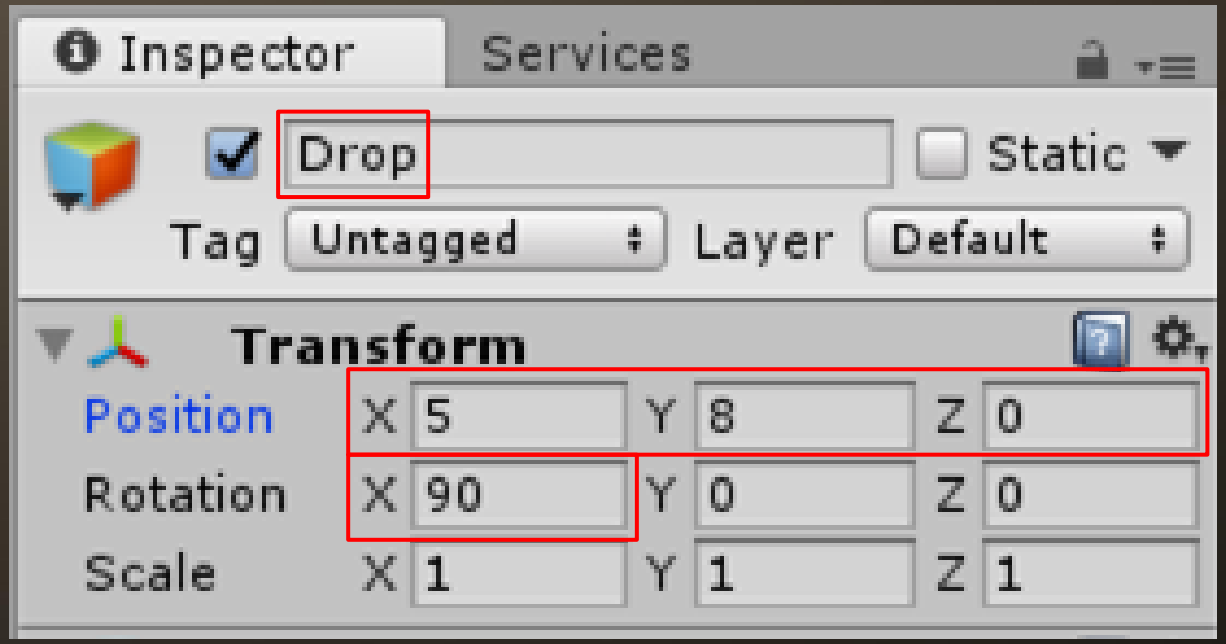
시도6: 다양한 종류의 벽



Try 7
위에서 떨어지는 장애물

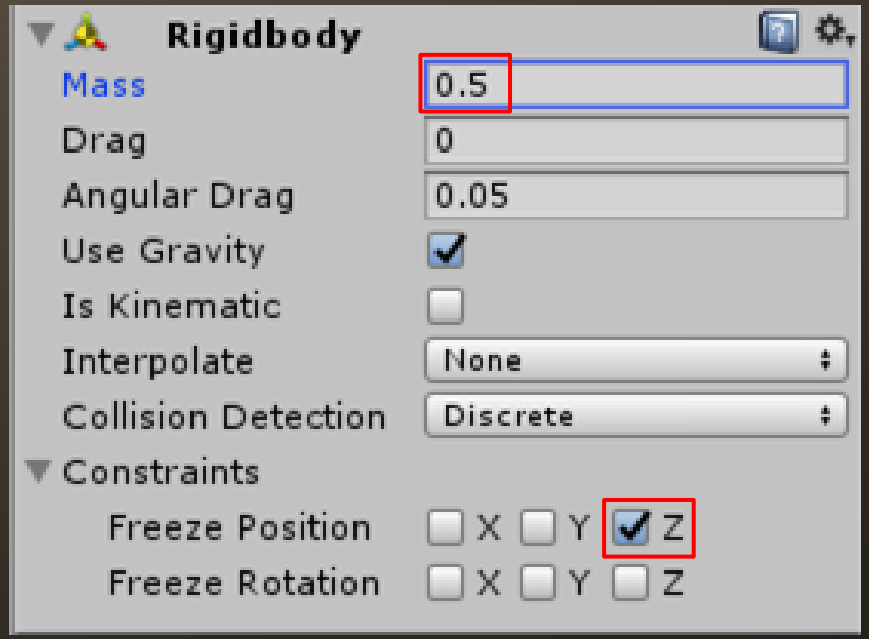
시도7: 위에서 떨어지는 장애물

- GameObject > 3D Object > Cylinder

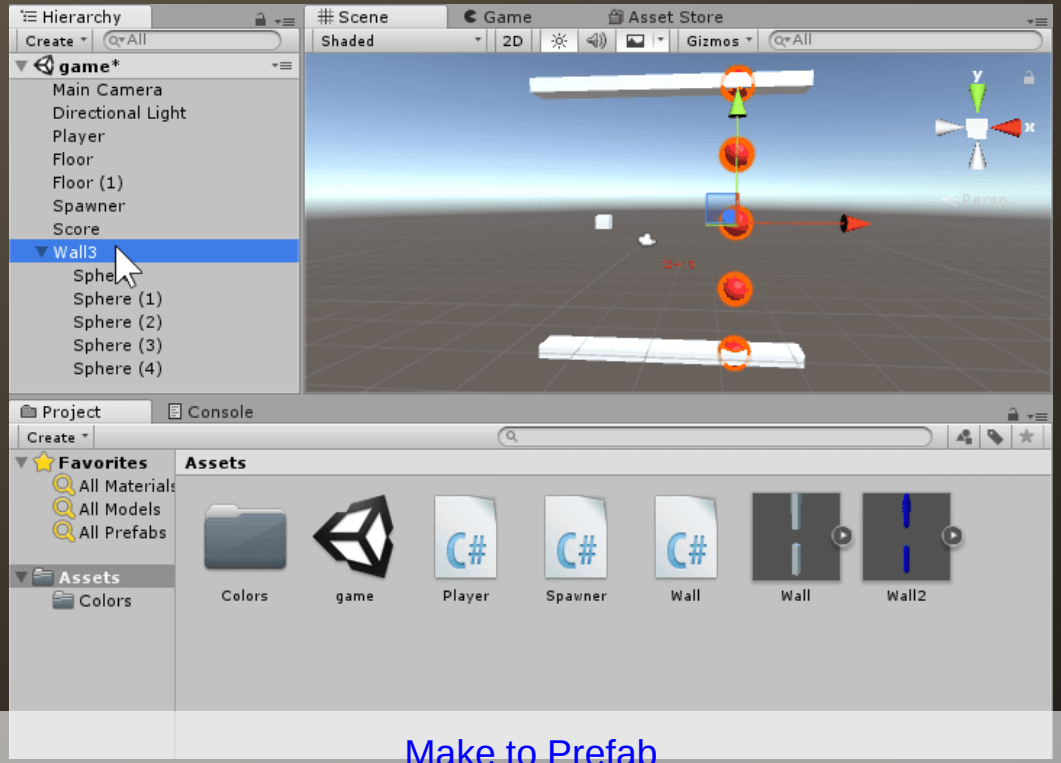


시도7: 위에서 떨어지는 장애물

- Component > Physics > Rigidbody

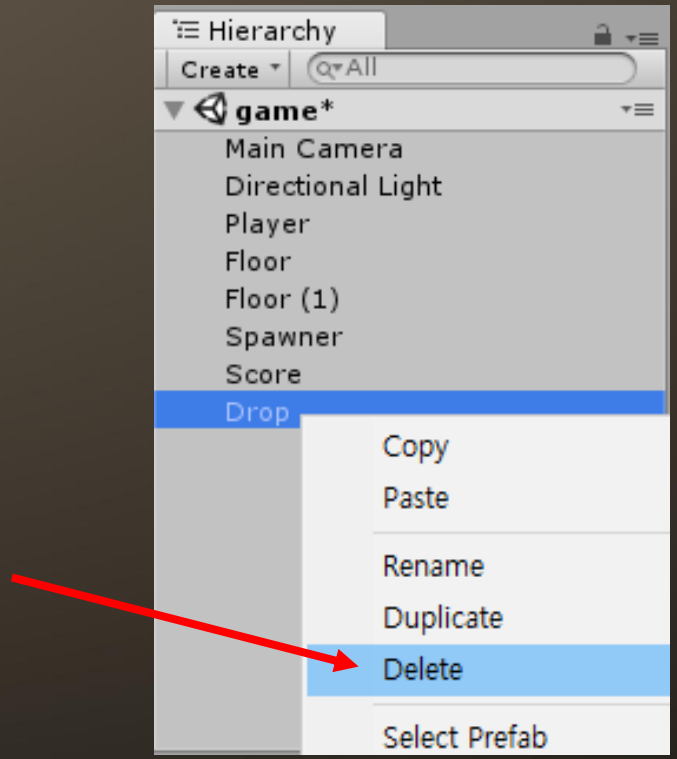


시도7: 위에서 떨어지는 장애물



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;
```

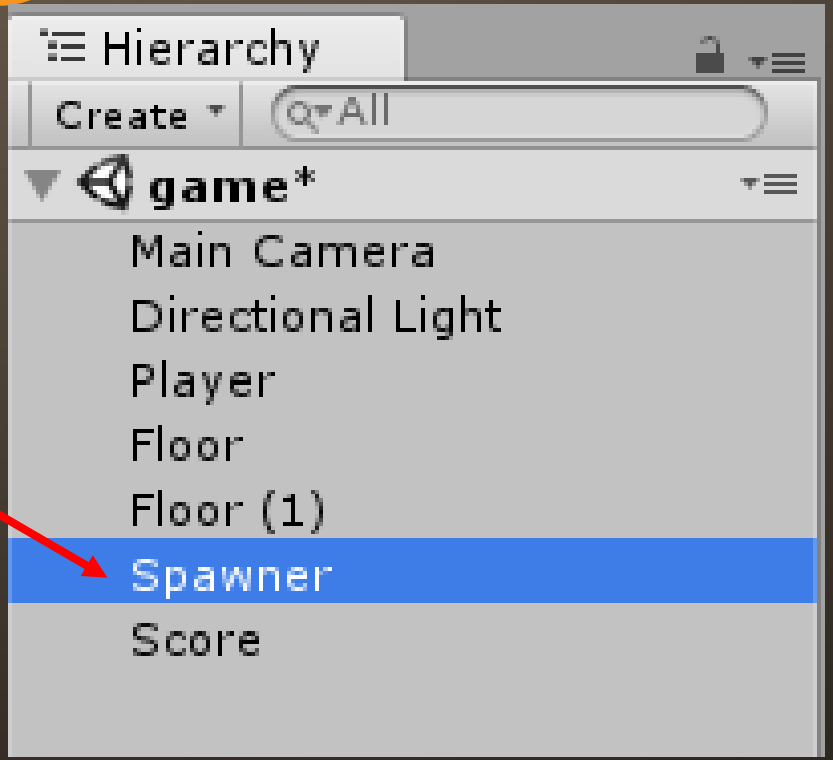
시도7: 위에서 떨어지는 장애물

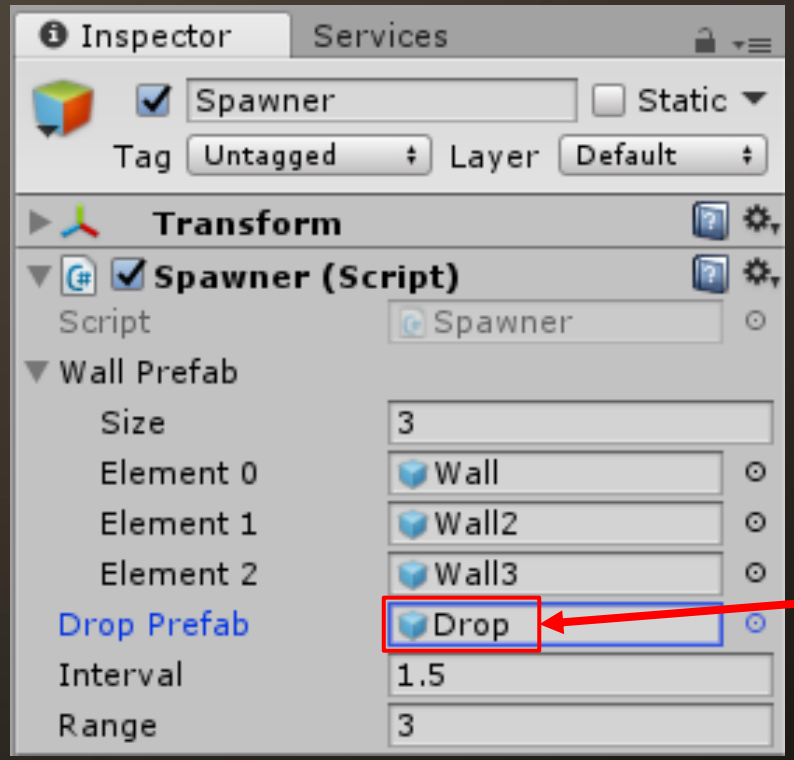
Spawner.cs

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