고주형

https://github.com/IamGroooooot/cien-unity-2019

2019/11/13


복습

Week 2

큐브 이동시키기

- Transform
- Rigidbody.Velocity
- 2가지 방법으로 큐브 이동
- Find("이름")
- Tag란?
- 게임 오브젝트들 구분하기
- Find한 오브젝트로 이동하기

총알 구현하기

- 총알 생성하기
- 총알의 충돌 감지
- 총알 종류 구분하기

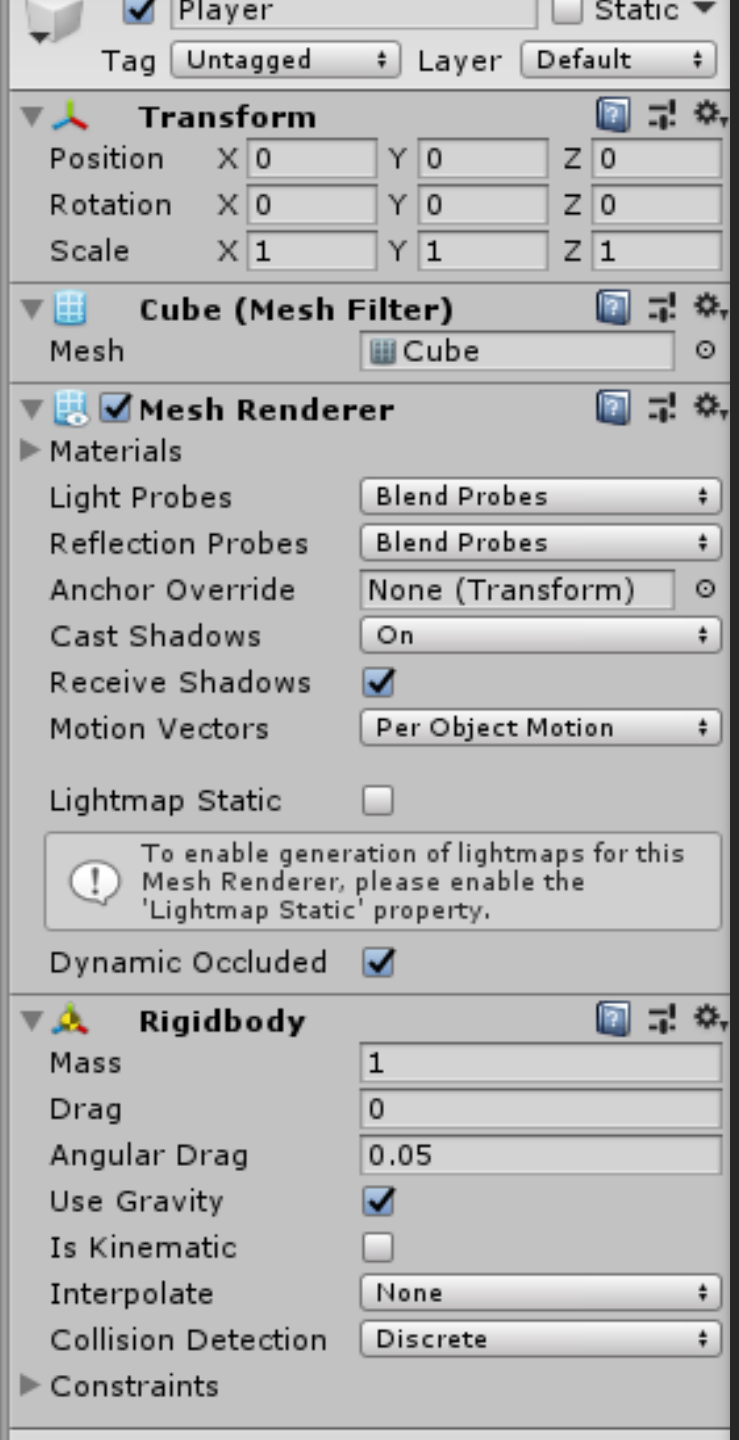
적 만들기

- 적 자동 사격
- Player 사망 구현
- 총알을 쏘면서 목표 지점까지 가기

2D 프로젝트는?

- 3D와 거의 비슷하다
- 나중에 시간이 된다면 …

Cube 이동시키기



Transform

→ 모든 오브젝트는 transform을 가짐

→ 오브젝트의 위치, 회전 그리고 스케일을 저장하고 다루기 위해서 사용합니다

여러가지 값들

- 위치
- 회전
- 크기

유니티의 2가지(Global/Local) 좌표
Transform의 Position과 LocalPosition

Position

절대적인 위치

절대적인 Global 좌표를 사용함
= World 기준의 좌표
= 원점(0, 0, 0)으로부터의 좌표
= 절대적이다 (상대적이지 않음)

인스펙터에서 보여주는
가장 최상위 부모의 위치는
절대위치이다

LocalPosition

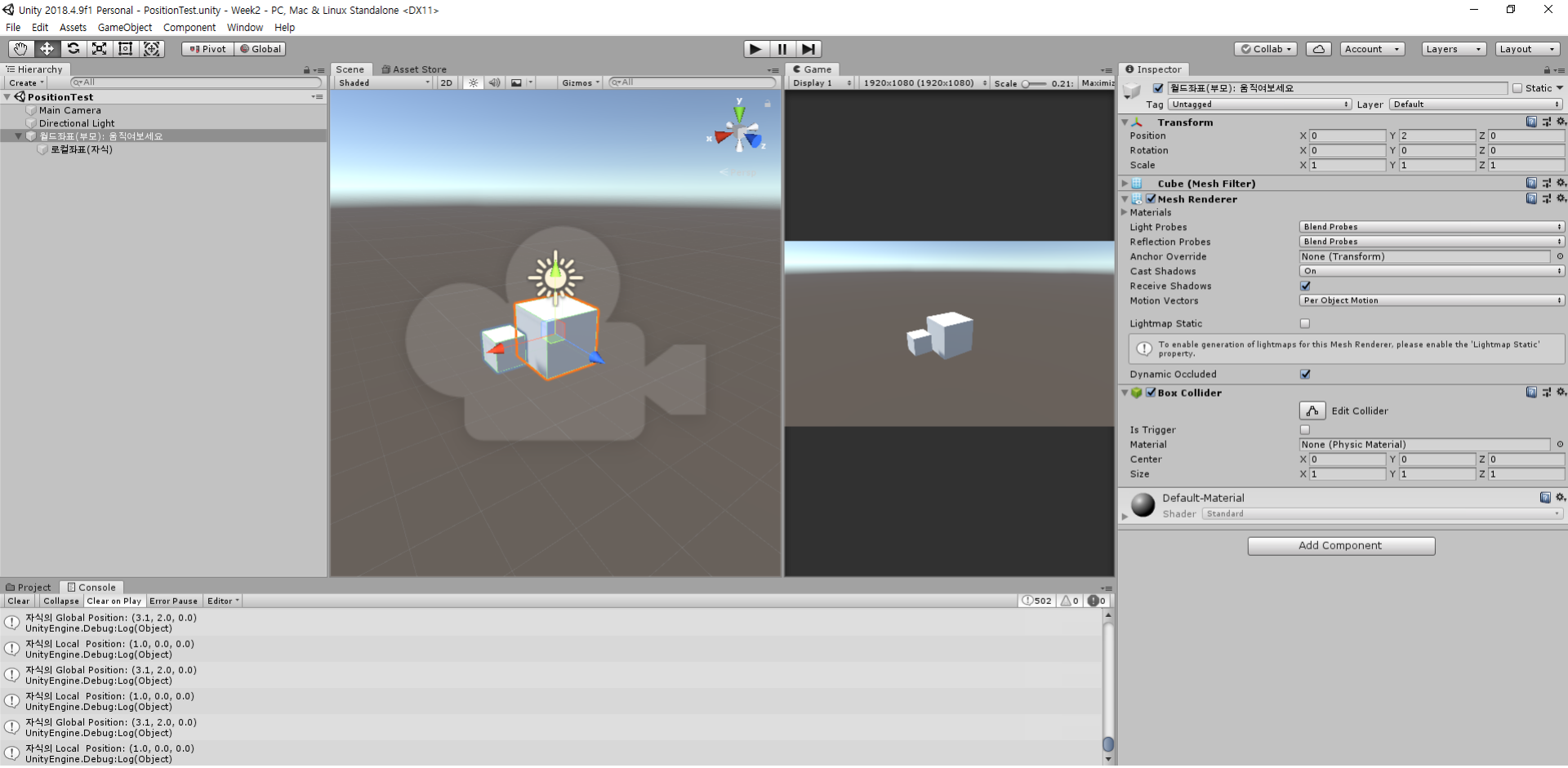
상대적인 위치

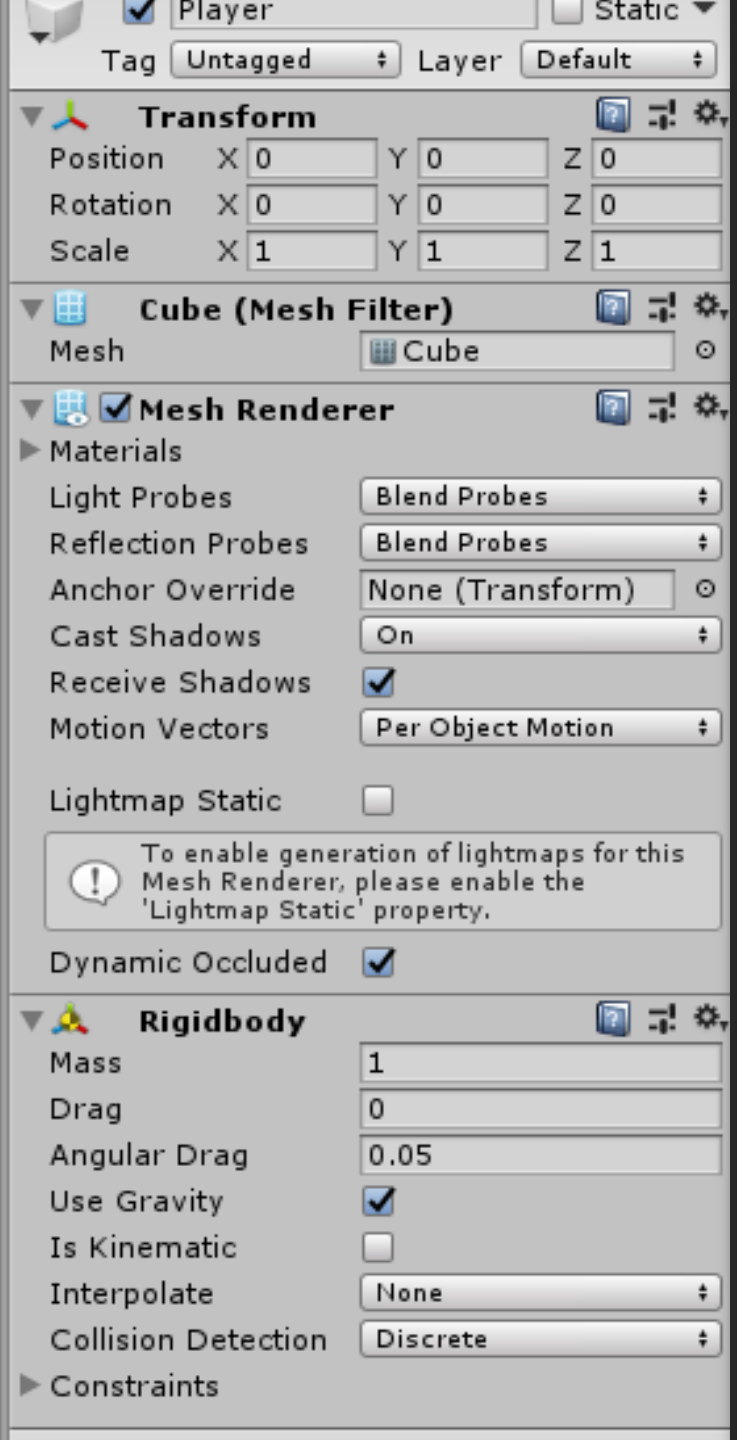
상대적인 Local 좌표를 사용함
= 부모 기준의 좌표
= 부모로부터의 좌표
= 상대적이다 (절대적이지 않음)

인스펙터에서 보여주는
모든 자식의 위치는
상대위치이다

Q. 부모를 움직이면 자식의 (인스펙터상) Position은 바뀔까요? 안 바뀔까요?

Debug를 찍어서 확인해보자
"PositionTest 씬 참고"



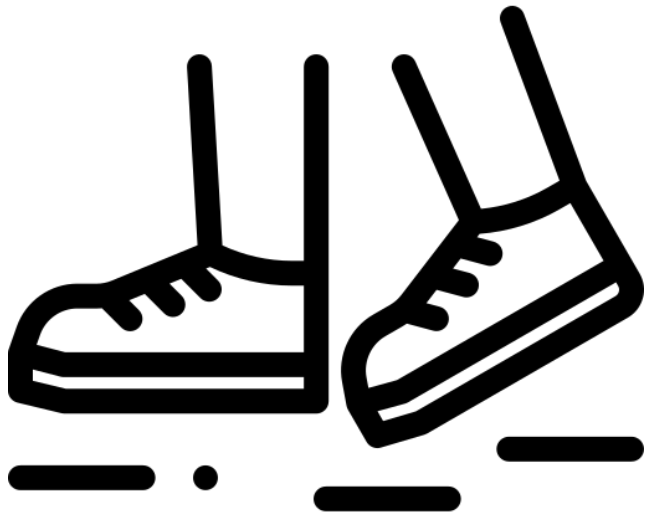


Rigidbody

→ GameObject가 물리 제어로 동작하게 합니다

여러가지 값들

- Mass: 질량
- Drag: 공기 저항
- Angular Drag: 토크로 회전할 때 공기 저항이 영향을 미치는 정도
- Use Gravity: 중력 작용 여부
- Is Kinematic: 물리 엔진으로 제어되지 않고 오로지 Transform 으로만 조작됩니다



3가지 방법으로 GameObject를 움직여보자

"MoveObjects 씬 참고"

1. Transform
   = 위치를 바꾼다
2. Rigidbody-Velocity
3. Rigidbody-Force

Transform의 Translate 사용

- Translate(벡터);
- 매 프레임마다
  벡터만큼 위치를 움직임

(주의)
상대좌표를 사용한다.
= 상대적으로 움직인다!
월드기준으로 이동시키려면 2번째 인자로 Space.World를 줘야한다

"TranslatePractice 씬 참고"

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

public class MoveByTranslate : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

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

Rigidbody의 Velocity 사용

- 게임오브젝트의 리지드바디 안에 있는
  velocity = 벡터;
- GetComponent<Rigidbody>().Velocity = 벡터;

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

public class MoveByVelocity : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        GetComponent<Rigidbody>().velocity = new Vector3(1f, 0,0);
    }

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

    }
}
```

Rigidbody의 Force 사용

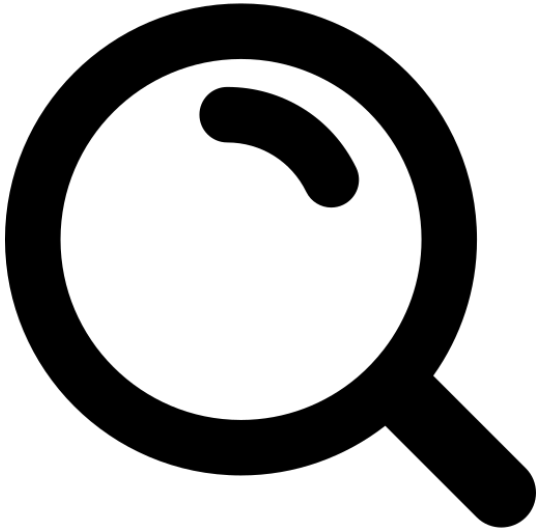
- 게임오브젝트의 리지드바디에 있는 AddForce함수 호출
- GetComponent<Rigidbody>().AddForce(벡터);
- 매 프레임마다 Force를 가함
- 가속도 생김!

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

public class MoveByForce : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        GetComponent<Rigidbody>().AddForce(new Vector3(1, 0,0));
    }
}
```

다른 게임 오브젝트에 접근하고 싶을 땐?

- (EX) 이름이 target인 게임 오브젝트를 찾아서 옮겨주고 싶다
- (EX) Player를 옮기고 싶다
- “FindTarget 씬 참고”

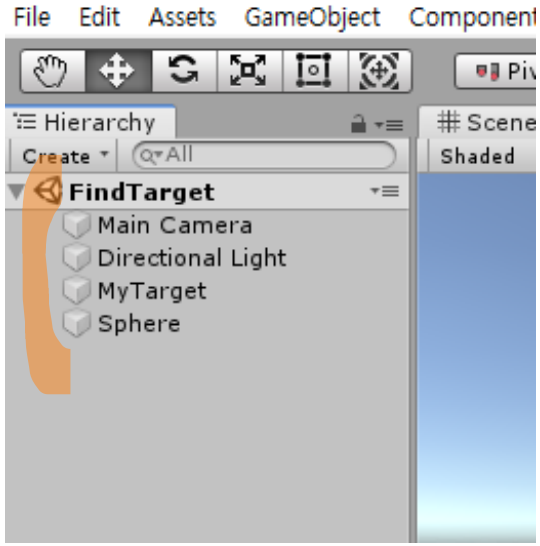
게임오브젝트를 찾는 방법 2가지



1. 게임오브젝트의
   이름(name)으로 찾기

2. 우리가 지어준
   별명(tag)으로 찾기

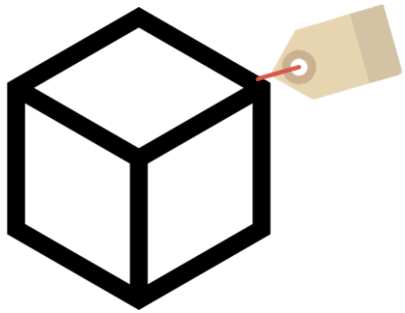
1. 이름으로 찾기



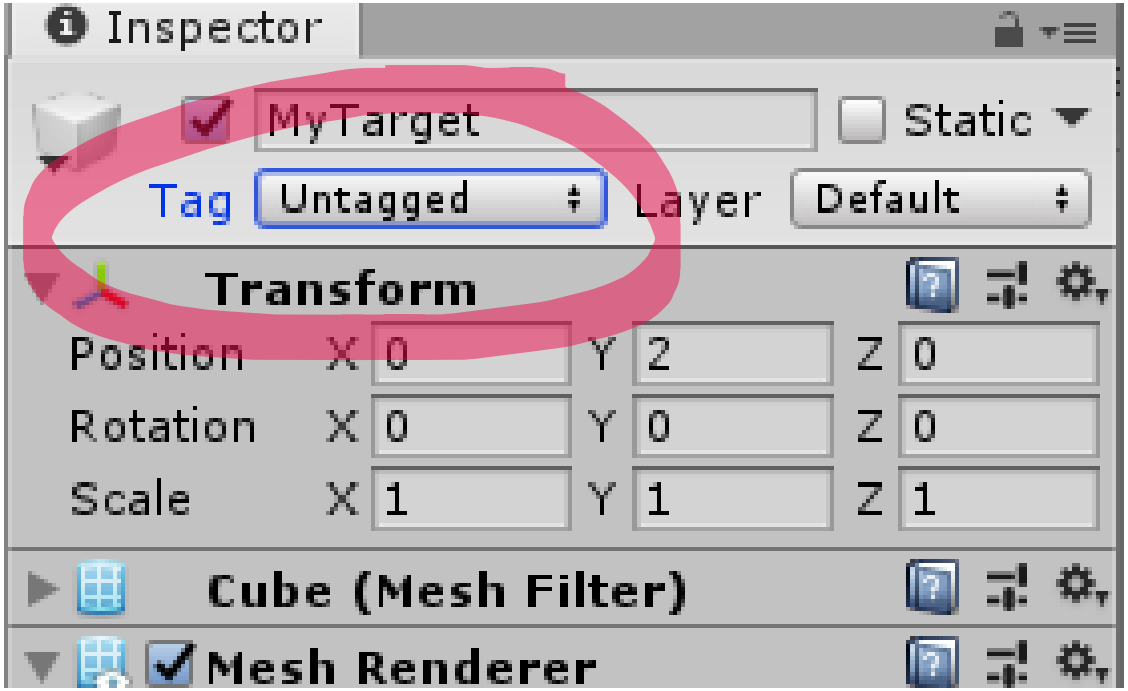
- Find("게임오브젝트_이름")
- GameObject Class에 정의된 Find함수!

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

public class FindWithName : MonoBehaviour
{
    private GameObject myTarget;

    // Start is called before the first frame update
    void Start()
    {
        myTarget = GameObject.Find("MyTarget");

        if(myTarget != null)
        {
            Debug.Log("GameObject("+myTarget.name+")을 "+"찾았습니다");
        }
        else
        {
            Debug.Log("해당하는 게임 오브젝트를 찾지 못했습니다");
        }
    }

    // Update is called once per frame
    void Update()
    {
        myTarget.GetComponent<Rigidbody>().AddForce(2,0,0);
    }
}
```

Tag란?

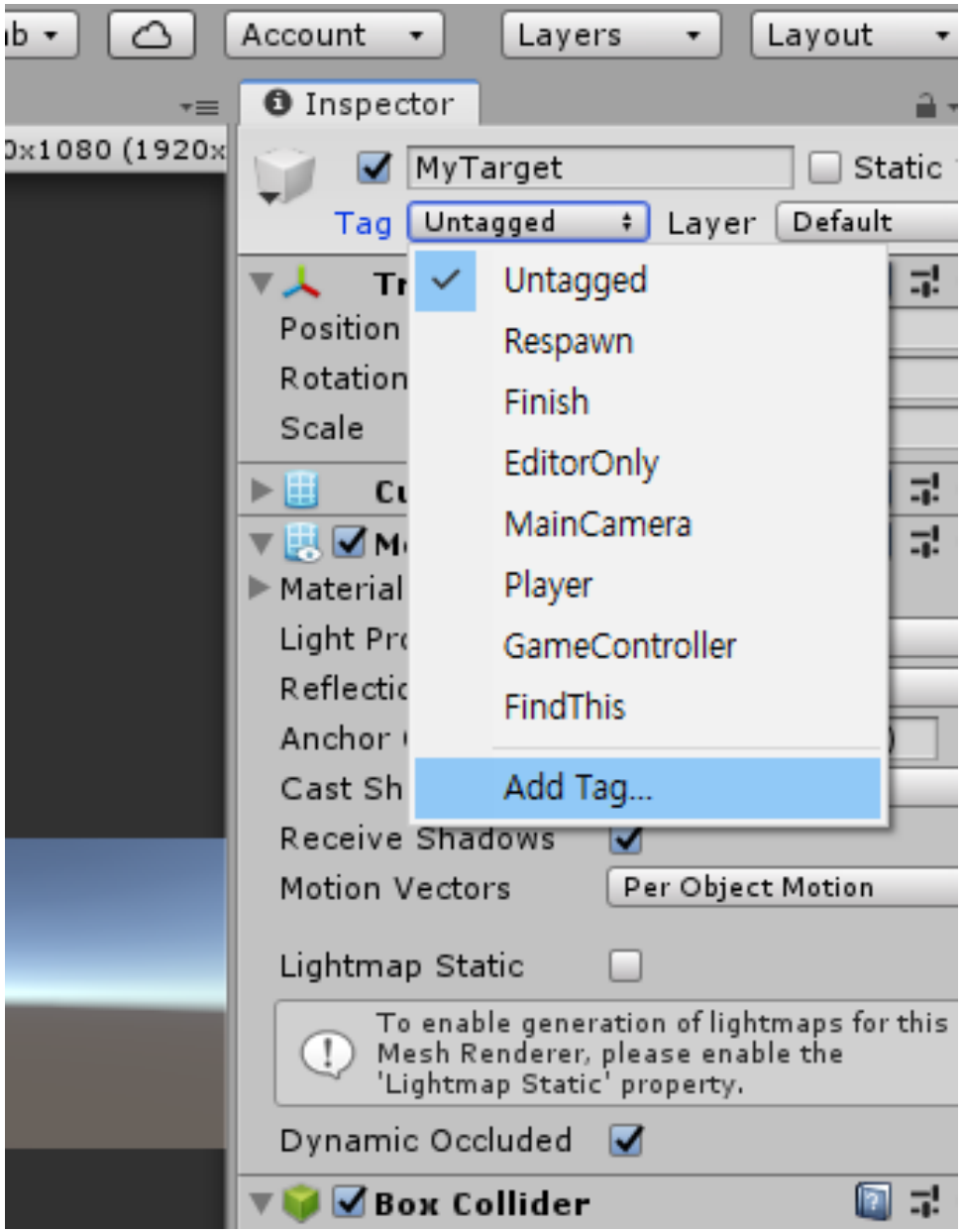


- 번역하면 꼬리표
- 별명(닉네임) 같은 것
- 우리는 게임 오브젝트에 별명을 지어줄 수도 있습니다!



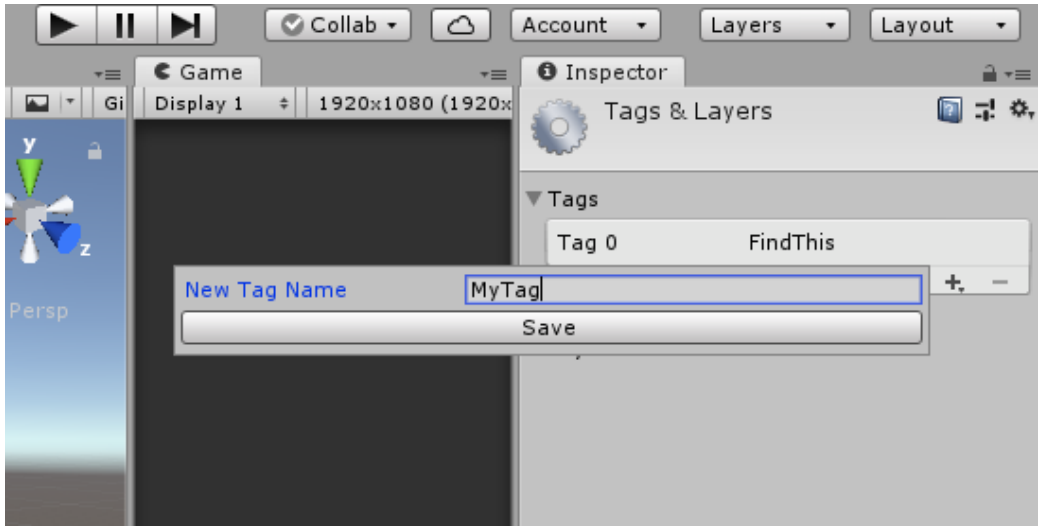
How to tag GameObject

Step 1
“Add Tag”



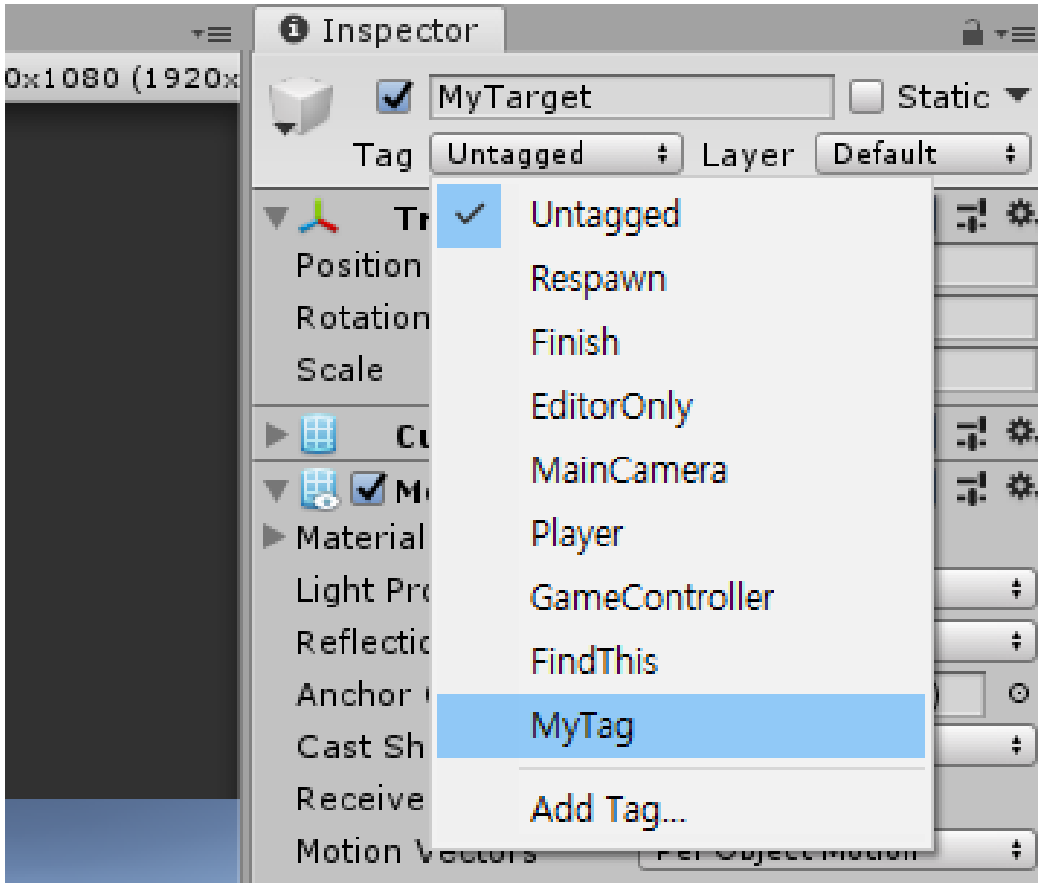
How to tag GameObject

Step 2
“Save”



How to tag GameObject

Step 3
“Set Tag”



2. Tag로 찾기

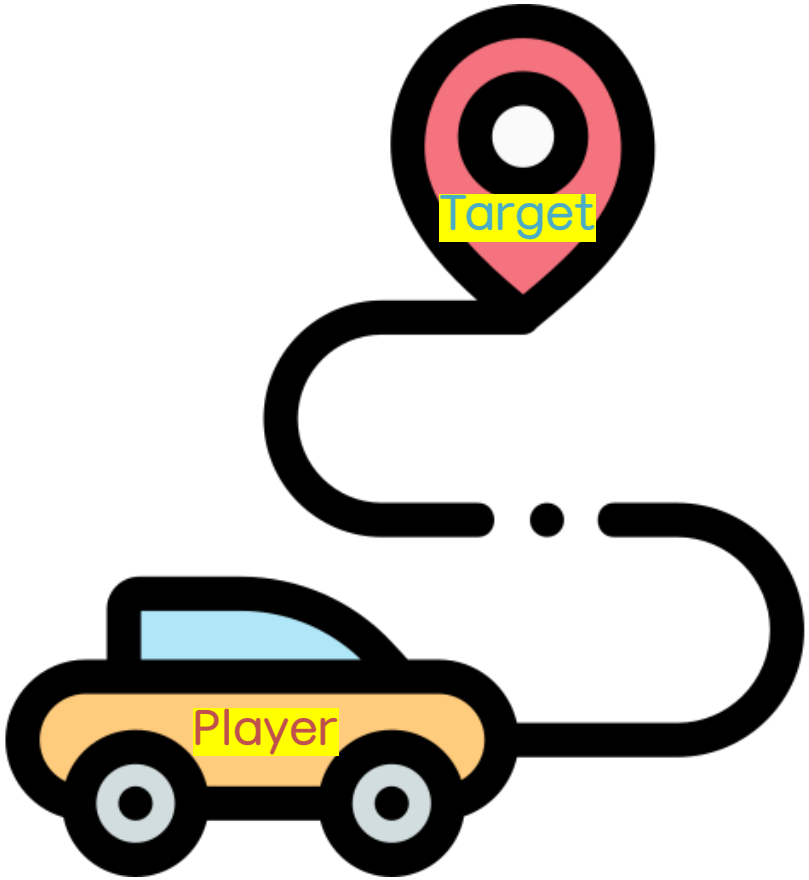
- 게임 오브젝트들의 이름이 겹칠 때 …
  ☞ 태그(Tag)로 찾자

- Tag를 달아서 게임 오브젝트를 그 태그로 찾자!
- GameObject Class에 정의된 FindGameObjectWithTag함수!

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

public class FindWithTag : MonoBehaviour
{
    private GameObject myTarget;

    // Start is called before the first frame update
    void Start()
    {
        myTarget = GameObject.FindGameObjectWithTag("MyTag");

        if (myTarget != null)
        {
            Debug.Log("GameObject(" + myTarget.name + ")을 " + "찾았습니다");
        }
        else
        {
            Debug.Log("해당하는 게임 오브젝트를 찾지 못했습니다");
        }
    }

    // Update is called once per frame
    void Update()
    {
        myTarget.GetComponent<Rigidbody>().AddForce(-2, 0, 0);
    }
}
```

Find한 게임오브젝트를 이용해보자

- Find한 게임오브젝트로 이동해보자



#총알 생성하기
#총알이 충돌했을 때
#Destroy(적)
#총알 구분하기

총알 구현하기

총알을 발사해서 적을 죽이는 것까지 구현해봅시다

발사할 총알을 만들자

- 먼저 총알을 만들어봅시다

1. 총알을 잘 만든다
2. 속도 부여

- 이 게임 오브젝트를 Prefab으로 만들자

 - Hierarchy창에서 Project창으로 끌어당기기 (Drag and Drop)

- Prefab이란? 저장해두는 GameObject

 - 삭제해도 Asset에 남아있다!!

총알 발사! <Step 1> "ShootBullet_1 씬 참고"

동적으로 게임 오브젝트를 생성하고 싶을 때

= 필요할 때 생성

= 스페이스바를 누를 때 생성

Instantiate함수를 쓰자

Instantiate(게임오브젝트)

Create GameObject

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

public class CreateBullet : MonoBehaviour
{
    public GameObject bulletPrefab;
    public Transform shootTranform;
    private Vector3 shootPosition;

    // Start is called before the first frame update
    void Start()
    {
        // bullet 생성 1 // 어떻게 생성? (회전도 크기도) bulletPrefab 그대로 //어디에 생성? bullet의 position 사용
        Instantiate(bulletPrefab);

        // bullet 생성 2 // 어떻게 생성? bulletPrefab의 rotation //어디에 생성? 게임오브젝트 ShootPosition의 위치
        shootPosition = new Vector3(shootTranform.position.x, shootTranform.position.y, shootTranform.position.z);
        Instantiate(bulletPrefab, shootPosition, bulletPrefab.transform.rotation);
    }

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

    }
}
```

총알 발사! <Step 2>

Space키를 누르면 총알 발사!!!

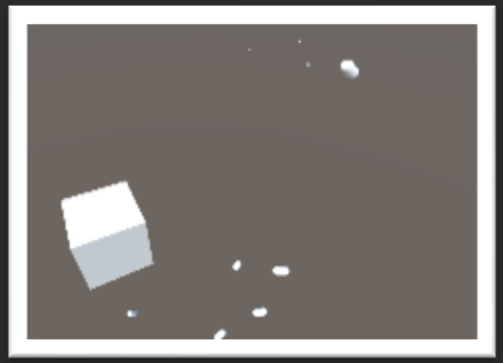
- Update에 넣으면 부왁!

키를 (누르고 있을 동안이 아니라) 누를 때만 발사하도록 해보자.

- GetKey? 누르고 있을 동안
- GetKeyDown? 누를 때
- GetKeyUp? 누르고 있다가 뗄 때

Just GetKey

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

public class CreateBullet_GetKey : MonoBehaviour
{

    public GameObject bulletPrefab;
    public Transform shootTranform;
    private Vector3 shootPosition;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.Space))
        {
            shootPosition = new Vector3(shootTranform.position.x, shootTranform.position.y,
 shootTranform.position.z);
            Instantiate(bulletPrefab, shootPosition, bulletPrefab.transform.rotation);
        }
    }
}
```



GetKeyDown

총알 발사! <Step 3>

Space를 계속 누르고 있으면 자동 발사하게 하고 싶다.

- 3초마다 자동 발사하는 것을 구현해봅시다

3초가 지났는지를 판별할 Timer가 필요하다

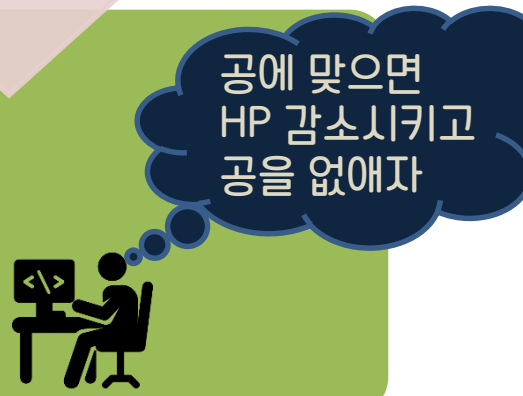
시간에 관한 것은 하나만 배웠다.

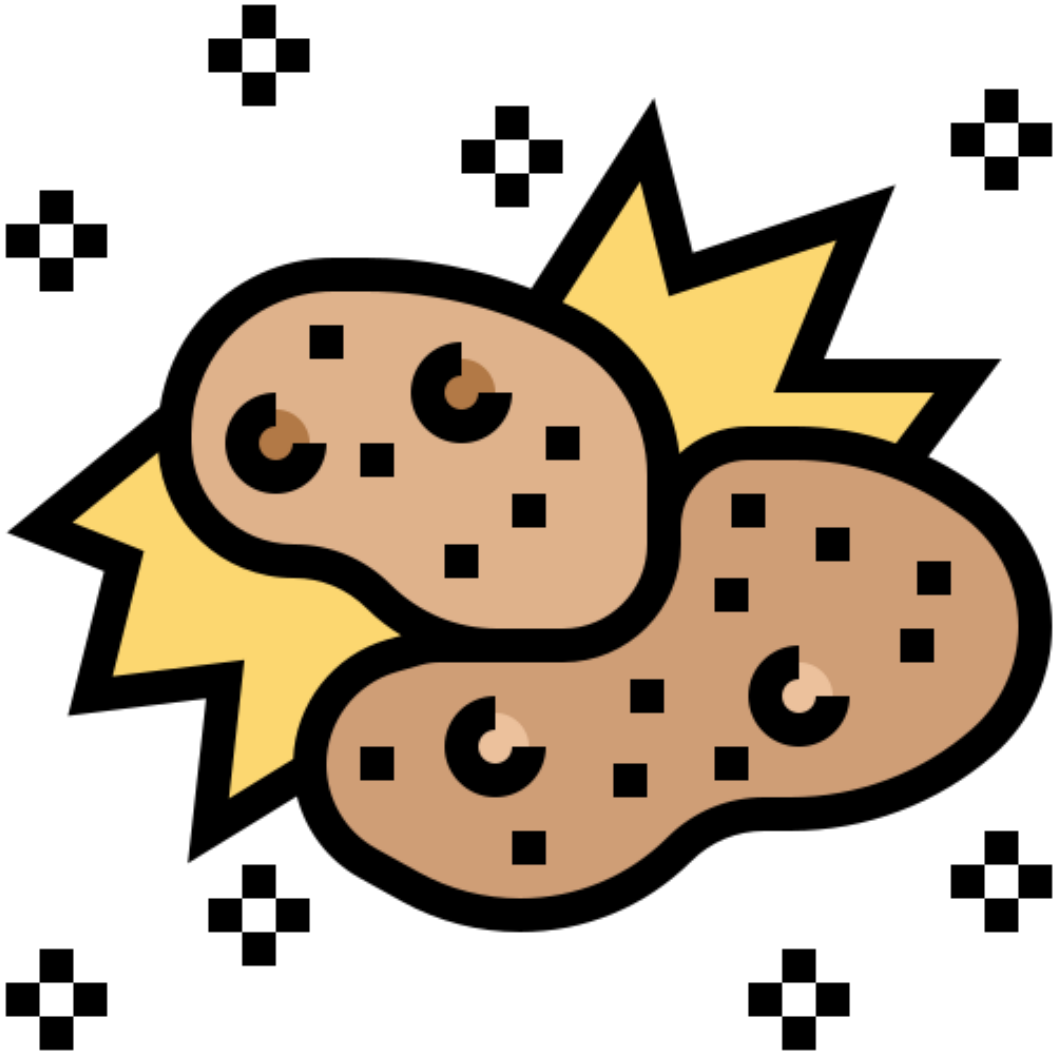
- 바로 Time.deltaTime!
- 이전 프레임이 걸린 시간!

총알이 충돌했을 때

적이 총알에 맞을 때 그 두개의 물체는 충돌했다고 합니다.

충돌한 게임 오브젝트들을 어떻게 인식하고 접근할 수 있을까요?





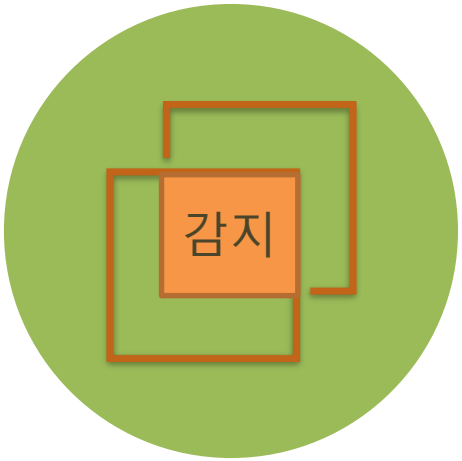
먼저,
유니티가
충돌을
어떻게
처리하는지
배워봅시다!

유니티의 충돌하기 위한 조건들

1. 두 GameObject 모두 Collider를 가지고 있어야 한다.

2. 둘 중 하나는 Rigidbody를 가지고 있어야 한다.
   (단, 한 GameObject가 고정되어 있다면 움직이는 쪽이 Rigidbody를 가지고 있어야 한다)

유니티의 충돌의 종류



COLLISION: 물리연산 하는 충돌

부딪칠 때!

TRIGGER: 물리연산 없는 충돌

감지할 때!

유니티 충돌: Collision

Collider를 달면 기본적으로 물리적인 충돌을 일으키면서 충돌을 감지한다.

예를 들어서,

- 물체 두개를 물리적으로 충돌시키고 싶다!

Collision 함수들

Collision Detection의 종류

- 충돌 시작할 때 (= 닿았을 때)
  - ☞ OnCollisionEnter 함수 호출
- 충돌이 지속되는 한
  - ☞ OnCollisionStay 함수 매 프레임마다 호출
- 충돌 끝날 때
  - ☞ OnCollisionExit 함수 호출

Collision 함수들

그럼 OnCollision****함수들은 어떻게 작동할까요?

BAAM!!!

유니티 충돌: Trigger

Collider를 감지용으로 사용하고 싶을 때

예를 들어서,

- 어떤 공간에 들어왔음을 감지하고 싶을 때
- 독 병에 근처에 가면 병을 줍는다!

Collider를 Trigger로 설정하기

- Trigger로 쓰려면 Collider의
  isTrigger 속성을 체크해야 합니다!

- (주의) isTrigger가 체크된 물체는
  다른 물체에 부딪히지 않습니다!
  다른 물체와 닿으면 그저 뚫고 지나갑니다!

Trigger 함수들

Trigger면 충돌했을 때
OnCollision****함수가 아니라
OnTrigger****함수를 호출합니다

Trigger Detection의 종류

- 충돌 시작할 때 (= 닿았을 때)
  ☞ OnTriggerEnter 함수 호출
- 충돌이 지속되는 한
  ☞ OnTriggerStay 함수 매 프레임마다 호출
- 충돌 끝날 때
  ☞ OnTriggerExit 함수 호출

Trigger 함수들

그럼 OnTrigger****함수들은 어떻게 작동할까요?

OnCollision****함수들과 마찬가지로 인자에 충돌한 애 던져줌

But Type이 달라!

Collider 형태로 줌ㅇㅇ

적은 어떻게 죽이나요?

충돌한 것까지 구현함 ㅇㅇ.
적은 어떻게 없앰?
Destroy함수 쓰면 ㅇㅋ!

Destroy(게임오브젝트)
 - 전달한 게임오브젝트를 없애는 함수

Destroy함수

Destroy함수는 유니티의 GameObject, Component, Asset을 없애는 함수입니다.

게임오브젝트를 삭제하고 싶으면 정확하게 없앨 게임오브젝트를 인자로 줘야 됩니다.

Destroy(this); // 의도한 코드인가요? 조심!

- 스크립트만 삭제. 게임오브젝트는 살아있음.

Destroy(this.gameObject);

- 이 스크립트를 달고 있는 게임오브젝트 삭제.

총알 완.벽. 구현 ㄱㄱ

Destroy함수를 써서 다른 게임오브젝트와 충돌하면 없애 버립시당

Destroy함수를 쓸 때 순서 주의!

1. Destroy(this.gameObject)
2. 이후 코드 실행 <- 실행 안됨

왜? 실행할 객체를 파괴했기 때문이다

총알이 충돌하는 게임오브젝트들을 구분해보자

- 지금 구현한 것은 부딪치기만 하면
  다 없애 버리는 코드입니다
- 어떻게 구분?
  - Tag달아서 구분하면 쉬움! (Tip) CompareTag("tag")
  - 다른 방법 쓰셔도 됩니당.

추가

- 총알이 멀리 가서 더 이상 필요 없을 때? 정리해주자
  - Destroy(target,5) //5초 후에 Destroy함
  - void OnBecameInvisible(){ } // 안보이게 되면 호출

#자동 사격

#Player 사망

적 만들기

적의 공격을 구현해보자

- 적 움직임 Script…
- 적 총알 Prefab
- 적 총알 Script
  - 충돌 시 어떻게…
  - Z축으로 전진… velocity로…? 뭘로…
  - 속도는 얼마…
- 총알 생성 Script

Player 사망을 구현해보자 (같이)
(Game Over)

- 게임오버 씬을 만들자!
- 사망씬으로 이동
  - SceneManagement.SceneManager.LoadScene함수

총알을 쏘면서 목표지점까지 가보자 (실습) (Game Clear)

- 게임 클리어 씬을 만들자

- 목표지점에 Collider설정(isTrigger 체크)
- 목표지점 Trigger에 Enter하면 게임클리어 씬으로 이동

시간이 된다면 …

2D Project?

출처

유니티 매뉴얼
https://docs.unity3d.com/Manual/index.html