

3D 칼 전투 게임

10218 노니콜라이

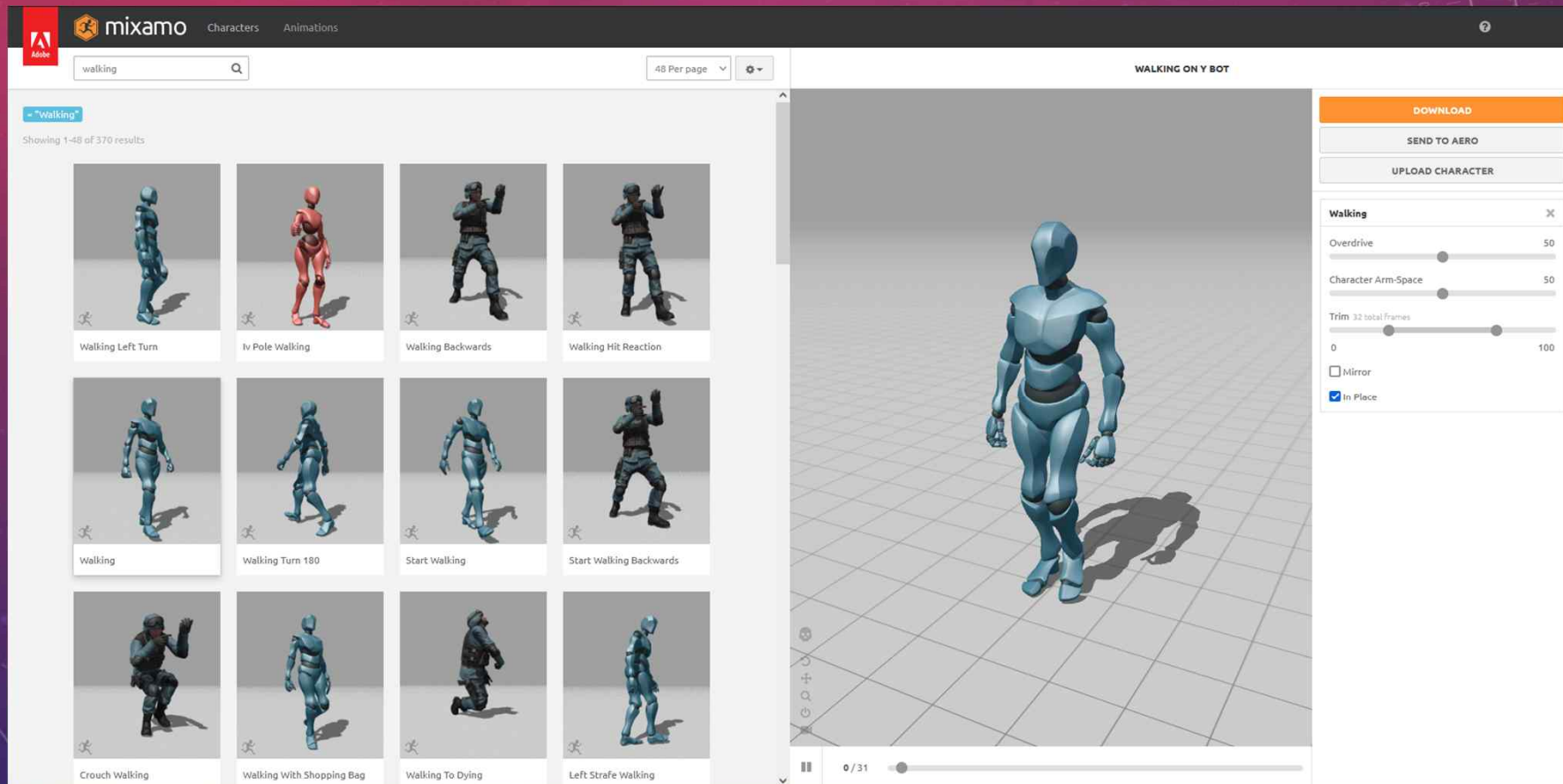
도구

1. Unity (게임 엔진)
2. Mixamo
3. Ezy-slice (유니티 메시 슬라이서 프레임워크)



Mixamo

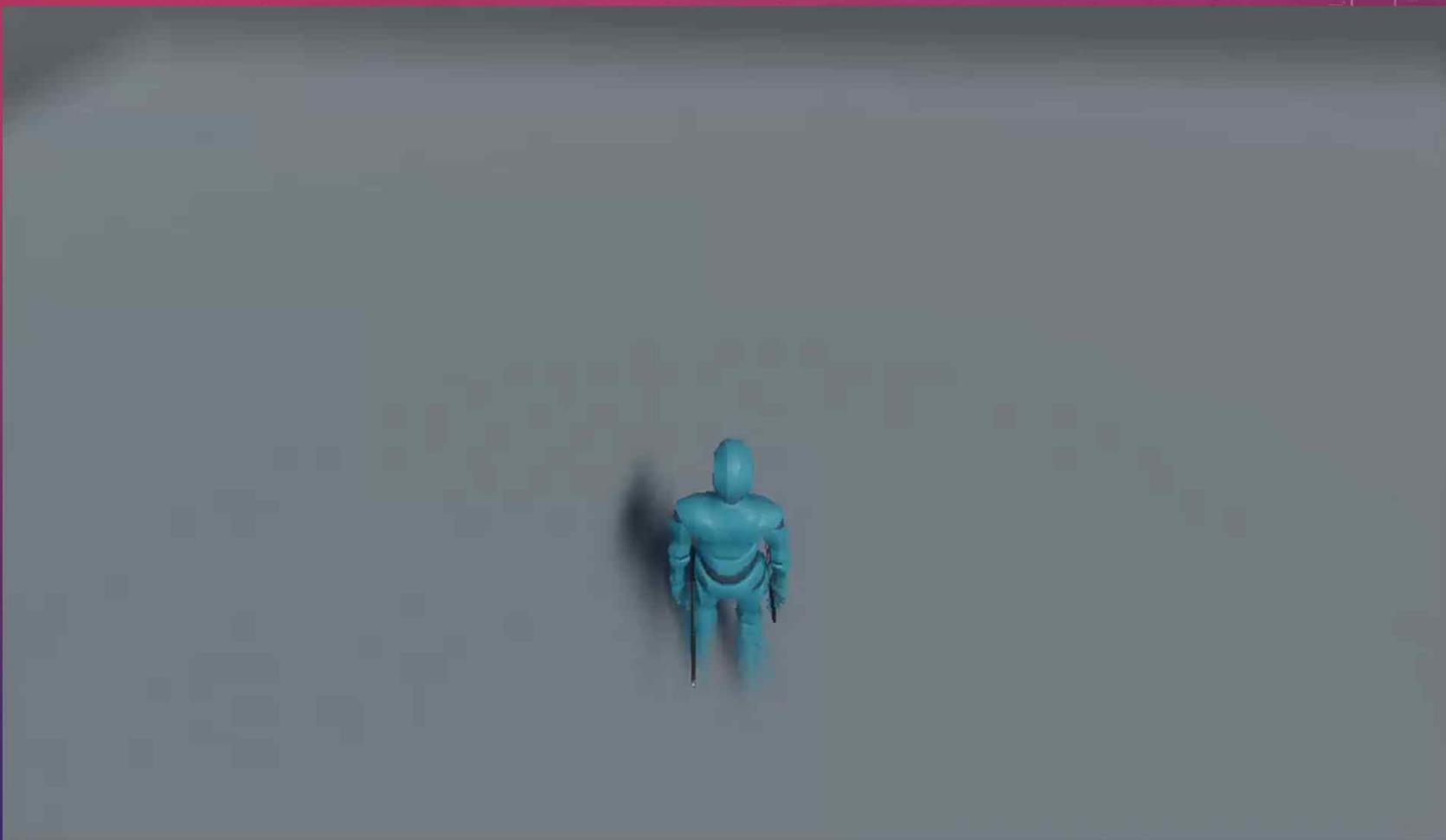
Mixamo는 많은 모델과 애니메이션이 있는 사이트입니다. 캐릭터 모델과 애니메이션은 이 사이트에서 가져왔습니다.



프로젝트 표시



카메라



카메라 스크립트

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
//카메라
☺ Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 1
public class CameraController : MonoBehaviour
{
    //최소 및 최대 각도(상하)
    public float minAngle;
    public float maxAngle;
    //마우스 감도
    public float lookSpeedX;
    public float lookSpeedY;
    float currentX = 0; //상하 회전
    public Transform pivot; //중심점
    public Transform target; //관찰대상(캐릭터)
    public CursorLockMode lockMode;
    ☺ Сообщение Unity | Ссылка: 0
    void Start()
    {
        Cursor.lockState = lockMode;
    }
    ☺ Сообщение Unity | Ссылка: 0
    void LateUpdate()
    {
        transform.position = target.position; //대상 위치 복사
        transform.rotation = Quaternion.Euler(0, transform.rotation.eulerAngles.y + Input.GetAxis("Mouse X") * lookSpeedX, 0); //좌우 회전

        //상하 회전
        currentX += lookSpeedY * Input.GetAxis("Mouse Y");
        currentX = Mathf.Clamp(currentX, minAngle, maxAngle);
        //중심점 회전
        pivot.rotation = Quaternion.Euler(currentX, pivot.rotation.eulerAngles.y, 0);
    }
}
```

변수

```
//최소 및 최대 각도(상하)
public float minAngle;
public float maxAngle;
//마우스 감도
public float lookSpeedX;
public float lookSpeedY;
float currentX = 0; //상하 회전
public Transform pivot; //중심점
public Transform target; //관찰대상(캐릭터)
```

▼ # ✓ Player (Script) ⓘ ⚙ ⋮

Script	Player	⊙
Anim	Idle (Animator)	⊙
Speed	3	
Cam	CameraController (Camera Controller)	⊙

카메라 업데이트

업데이트는 프레임당 한 번 호출됩니다.

Сообщение Unity | Ссылка: 0

```
void LateUpdate()
```

```
{  
    transform.position = target.position; //대상 위치 복사  
    transform.rotation = Quaternion.Euler(0, transform.rotation.eulerAngles.y + Input.GetAxis("Mouse X") * LookSpeedX, 0); //좌우 회전  
    currentX += lookSpeedY * Input.GetAxis("Mouse Y"); //상하 회전  
    currentX = Mathf.Clamp(currentX, minAngle, maxAngle); //보기 각도 제한  
    //중심점 회전  
    pivot.rotation = Quaternion.Euler(currentX, pivot.rotation.eulerAngles.y, 0);  
}
```


캐릭터

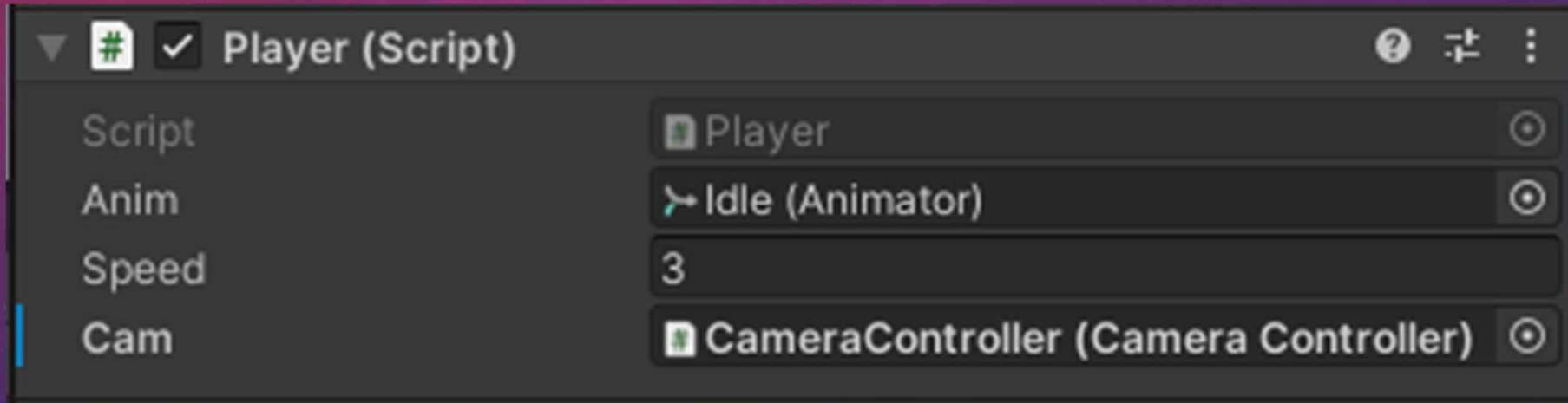


캐릭터 스크립트

```
using UnityEngine;
//캐릭터
// 스크립트 Unity (소스링크: 2) | 소스링크: 0
public class Player : MonoBehaviour
{
    Rigidbody rb; //강제(물리력)
    public Animator anim; //애니메이션
    public float speed; //캐릭터의 속도
    public CameraController cam; //카메라
    bool waving = false; //캐릭터가 파동 공격을 하고 있습니까?
    //첫 번째 프레임 업데이트 전에 시작이 호출됩니다.
    // Сообщение Unity | 소스링크: 0
    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }
    // 업데이트는 프레임당 한 번 호출됩니다.
    // Сообщение Unity | 소스링크: 0
    void Update()
    {
        //Q 키를 누르지 않으면 캐릭터가 카메라 회전을 따라갑니다.
        if (!Input.GetKey(KeyCode.Q))
        {
            transform.rotation = cam.transform.rotation;
        }
        //사용자가 캐릭터 애니메이션을 위해 이동 버튼을 누르는지 확인합니다.
        // "State"가 0이면 서 있는 애니메이션이 재생되고, 1이면 걷는 애니메이션이 재생됩니다.
        if (Input.GetButton("Horizontal") || Input.GetButton("Vertical"))
        {
            anim.SetInteger("State", 1);
        }
        else
        {
            anim.SetInteger("State", 0);
        }
        // 파동 공격 중지
        if (Input.GetKeyUp(KeyCode.R)) //키를 누르지 않은 경우
        {
            anim.CrossFade("Idle", 0.25f);
            waving = false;
        }
        if (waving) return; //웨이브 공격을 하는 동안 아무것도 하지 않습니다.
        Vector3 newVel = transform.forward * Input.GetAxis("Vertical"); //앞뒤로 이동
        newVel += transform.right * Input.GetAxis("Horizontal"); //좌우 이동
        newVel = Vector3.ClampMagnitude(newVel, 1f); //대각선 속도 제한
        newVel *= speed; //속도
        newVel.y = rb.velocity.y; //중력에 대해 y축을 따라 동일한 속도를 유지합니다.
        rb.velocity = newVel;
        //사용자가 E 키를 누르면 기본 공격을 합니다.
        if (Input.GetKeyDown(KeyCode.E))
        {
            anim.CrossFade("Slash", 0); //모든 애니메이션에서 특정 애니메이션으로 이동합니다.
        }
        //사용자가 R 키를 누르면 파동 공격을 합니다.
        else if (Input.GetKeyDown(KeyCode.R))
        {
            anim.CrossFade("SwordWave", 0);
            waving = true;
        }
    }
}
```

캐릭터 변수

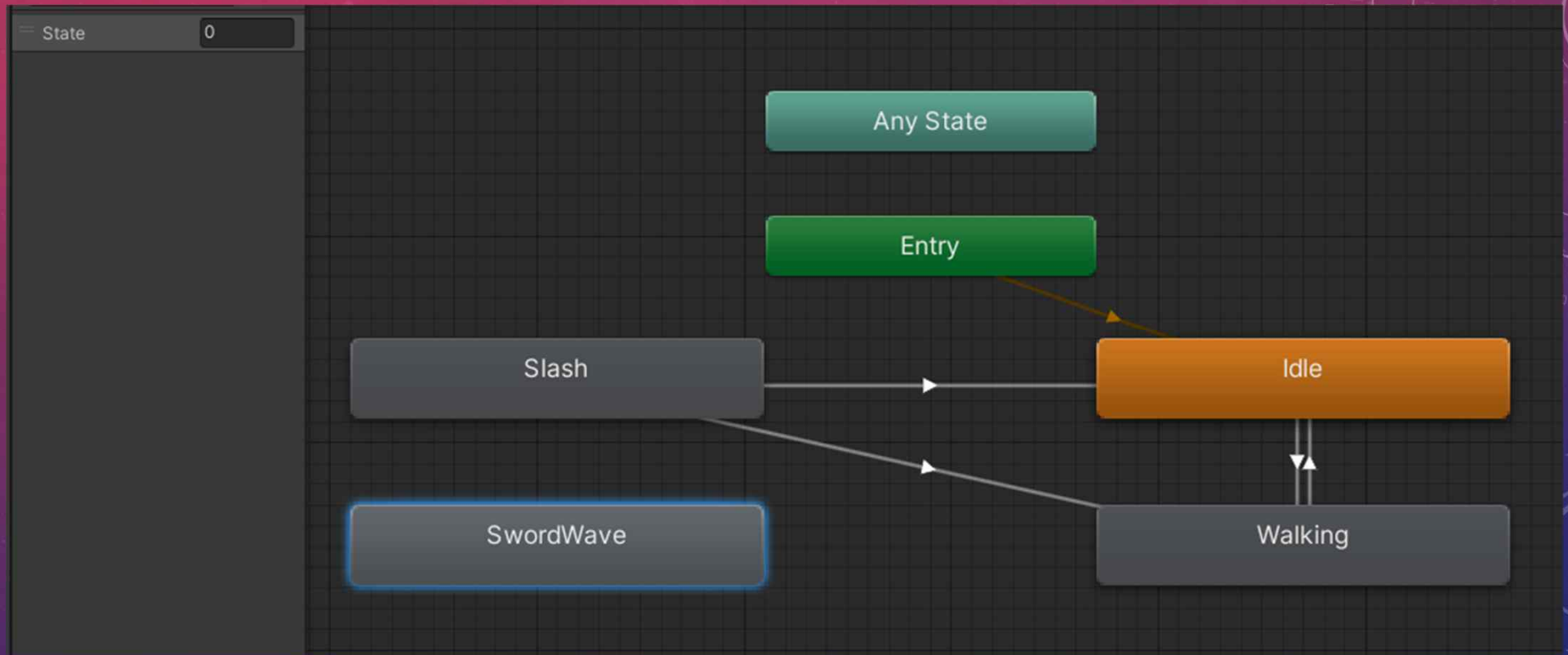
```
Rigidbody rb; //강체(물리학)  
public Animator anim; //애니메이션  
public float speed; //캐릭터의 속도  
public CameraController cam; //카메라  
bool waving = false; //캐릭터 파동 공격할 때 참입니다.
```



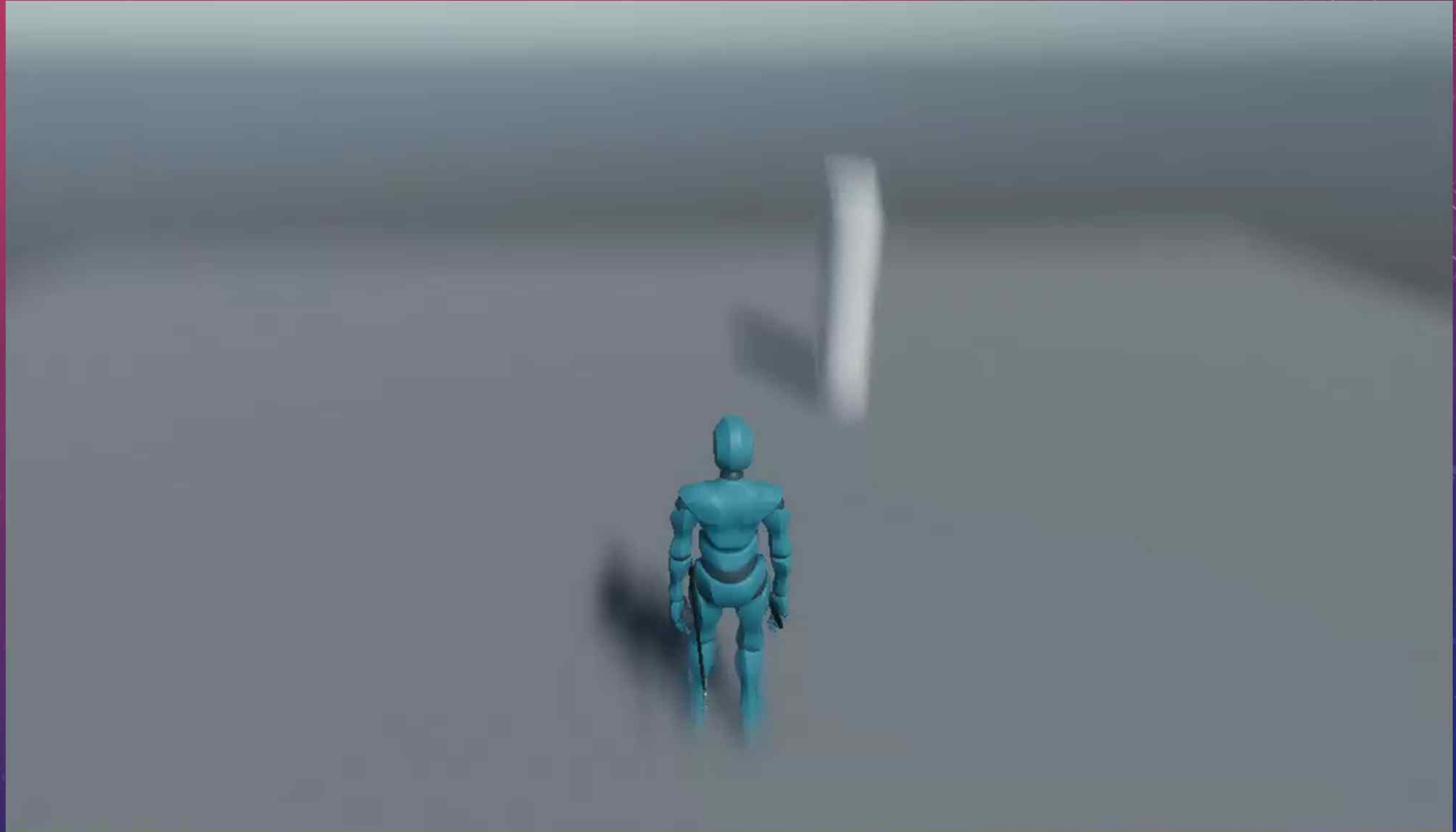
캐릭터 업데이트

```
// 업데이트는 프레임당 한 번 호출됩니다.
@ Сообщение Unity | Ссылка: 0
void Update()
{
    //Q 키를 누르지 않으면 캐릭터가 카메라 회전을 따라갑니다.
    if(!Input.GetKey(KeyCode.Q))
    {
        transform.rotation = cam.transform.rotation;
    }
    //사용자가 캐릭터 애니메이션을 위해 이동 버튼을 누르는지 확인합니다.
    //"State"가 0이면 서 있는 애니메이션이 재생되고, 1이면 걷는 애니메이션이 재생됩니다.
    if (Input.GetButton("Horizontal") || Input.GetButton("Vertical"))
    {
        anim.SetInteger("State", 1);
    }
    else
    {
        anim.SetInteger("State", 0);
    }
    // 파동 공격 중지
    if (Input.GetKeyUp(KeyCode.R))//키를 누르지 않은 경우
    {
        anim.CrossFade("Idle", 0.25f);
        waving = false;
    }
    if (waving) return;//웨이브 공격을 하는 동안 아무것도 하지 않습니다.
    Vector3 newVel = transform.forward * Input.GetAxis("Vertical");//앞뒤로 이동
    newVel += transform.right * Input.GetAxis("Horizontal");//좌우 이동
    newVel = Vector3.ClampMagnitude(newVel, 1f);//대각선 속도 제한
    newVel *= speed;//속도
    newVel.y = rb.velocity.y;//중력에 대해 y축을 따라 동일한 속도를 유지합니다.
    rb.velocity = newVel;
    //사용자가 E 키를 누르면 기본 공격을 합니다.
    if (Input.GetKeyDown(KeyCode.E))
    {
        anim.CrossFade("Slash", 0);//모든 애니메이션에서 특정 애니메이션으로 이동합니다.
    }
    //사용자가 R 키를 누르면 파동 공격을 합니다.
    else if (Input.GetKeyDown(KeyCode.R))
    {
        anim.CrossFade("SwordWave", 0);
        waving = true;
    }
}
```

캐릭터 애니메이션



기본 공격



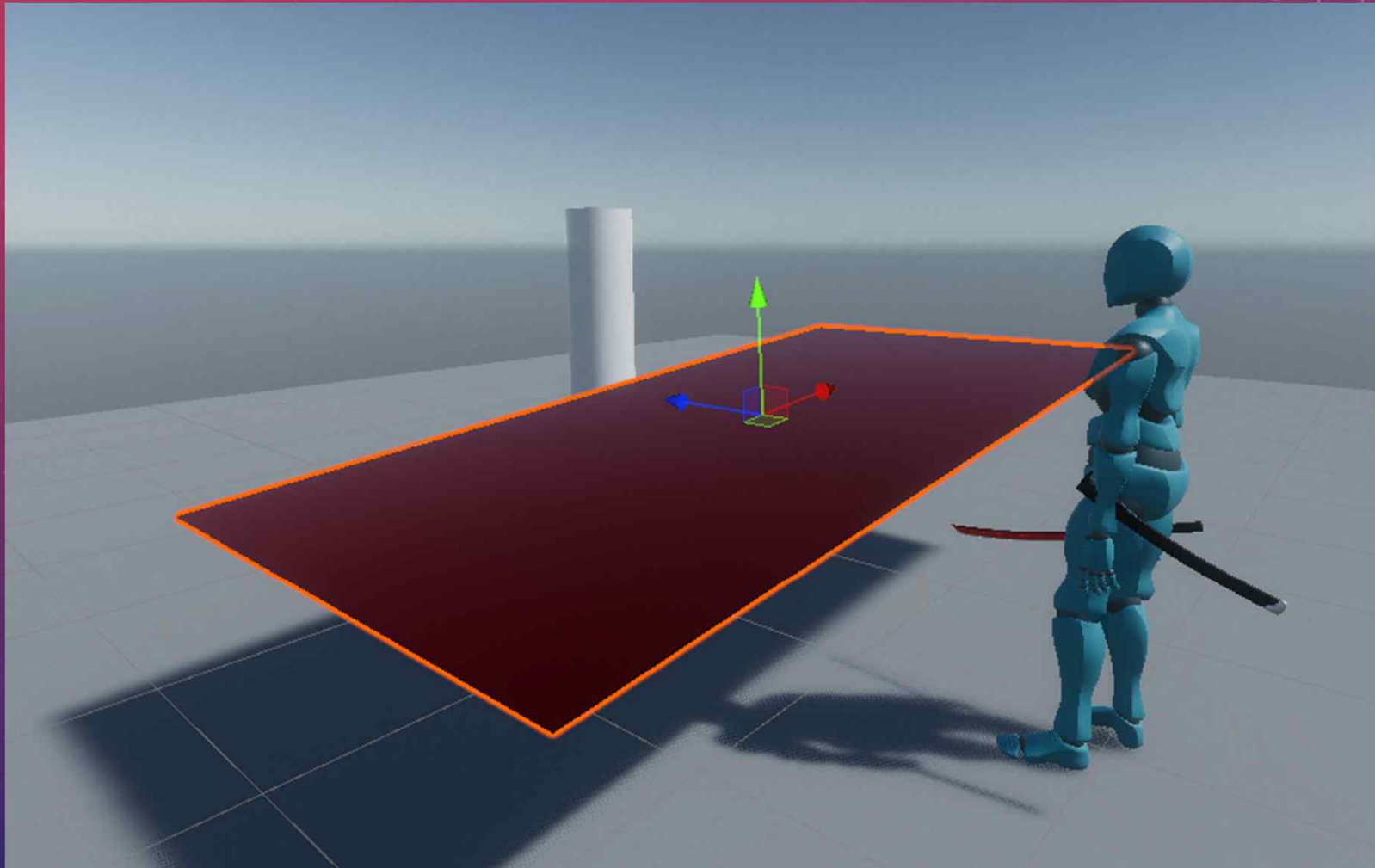
기본 공격 스크립트

```
using UnityEngine;
//기본 공격
public class KatanaSlice : MonoBehaviour
{
    public Transform cutPlane; //히트박스 위치
    public LayerMask layerMask; //슬라이스 가능한 객체만 감지하는 데 필요합니다.
    public float sliceX; //너비
    public float sliceZ; //길이
    //기본 공격 함수(애니메이션에서 호출)
    public void Slice()
    {
        //범위에 있는 슬라이스 가능한 객체만 가져오기
        Collider[] hits = Physics.OverlapBox(cutPlane.position, new Vector3(sliceX, 0.1f, sliceZ), cutPlane.rotation, layerMask);
        //각 개체를 슬라이스
        for (int i = 0; i < hits.Length; i++)
        {
            ObjectSlicer.Instance.Slice(cutPlane.position, cutPlane.up, hits[i].gameObject);
        }
    }
}
```

Katana Slice (Script)

Script	KatanaSlice
Cut Plane	Plane (Transform)
Layer Mask	Sliceable
Slice X	2
Slice Z	1

기본 공격 히트박스 표시



객체 슬라이서 스크립트

```
using EzySlice;
using UnityEngine;
// Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 3
public class ObjectSlicer : MonoBehaviour
{
    public Material crossMaterial; // 슬라이스 마크의 텍스처
    public float force = 100;
    public PhysicMaterial physMat; // 마찰, 탄력
    public static ObjectSlicer Instance; // 싱글톤
    // Сообщение Unity | Ссылка: 0
    public void Awake()
    {
        Instance = this; // 싱글톤
    }
    // 기본 컴포넌트들을 적용합니다.
    // Ссылка: 2
    public void AddHullComponents(GameObject go)
    {
        go.layer = 10;
        go.AddComponent<Hull>();
        Rigidbody rb = go.AddComponent<Rigidbody>();
        rb.interpolation = RigidbodyInterpolation.Interpolate;
        MeshCollider collider = go.AddComponent<MeshCollider>();
        collider.convex = true;
        collider.material = physMat;
        rb.AddExplosionForce(force, go.transform.position, 20); // 물체를 자를 때 힘을 가합니다.
    }
    // Ссылка: 2
    public (GameObject bottom, GameObject top) Slice(Vector3 pos, Vector3 dir, GameObject obj)
    {
        SlicedHull hull = obj.Slice(pos, dir, crossMaterial); // EzySlice 프레임워크의 함수
        // 슬라이스가 잘못된 경우 hull은 null입니다.
        if (hull != null)
        {
            GameObject bottom = hull.CreateLowerHull(obj, crossMaterial); // 윗부분 만들기
            GameObject top = hull.CreateUpperHull(obj, crossMaterial); // 아랫부분 만들기
            AddHullComponents(bottom);
            AddHullComponents(top);
            Destroy(obj); // 원래 개체를 파괴
            return (bottom, top);
        }
        return (null, null);
    }
}
```

슬라이서 변수

```
public Material crossMaterial; //슬라이스 마크의 텍스처  
public float force = 100;  
public PhysicMaterial physMat; //마찰, 탄력  
public static ObjectSlicer Instance; //싱글톤
```

▼ # Object Slicer (Script) ? 구 ⋮

Script	ObjectSlicer	⊙
Cross Material	RedGlow	⊙
Force	100	
Phys Mat	NoFriction	⊙

슬라이스 함수

Ссылка: 2

```
public (GameObject bottom,GameObject top) Slice(Vector3 pos, Vector3 dir, GameObject obj)
{
    SlicedHull hull = obj.Slice(pos, dir, crossMaterial);//EzySlice 프레임워크의 함수
    //슬라이스가 잘못된 경우 hull은 null입니다.
    if (hull != null)
    {
        GameObject bottom = hull.CreateLowerHull(obj, crossMaterial);//윗부분 만들기
        GameObject top = hull.CreateUpperHull(obj, crossMaterial);//아랫부분 만들기
        AddHullComponents(bottom);
        AddHullComponents(top);
        Destroy(obj);//원래 개체를 파괴
        return (bottom, top);
    }
    return (null, null);
}
```

위아랫부분 컴포넌트

//기본 컴포넌트들을 적용합니다.

Ссылка: 2

```
public void AddHullComponents(GameObject go)
```

```
{
```

```
    go.layer = 10;
```

```
    go.AddComponent<Hull>();
```

```
    Rigidbody rb = go.AddComponent<Rigidbody>();
```

```
    rb.interpolation = RigidbodyInterpolation.Interpolate;
```

```
    MeshCollider collider = go.AddComponent<MeshCollider>();
```

```
    collider.convex = true;
```

```
    collider.material = physMat;
```

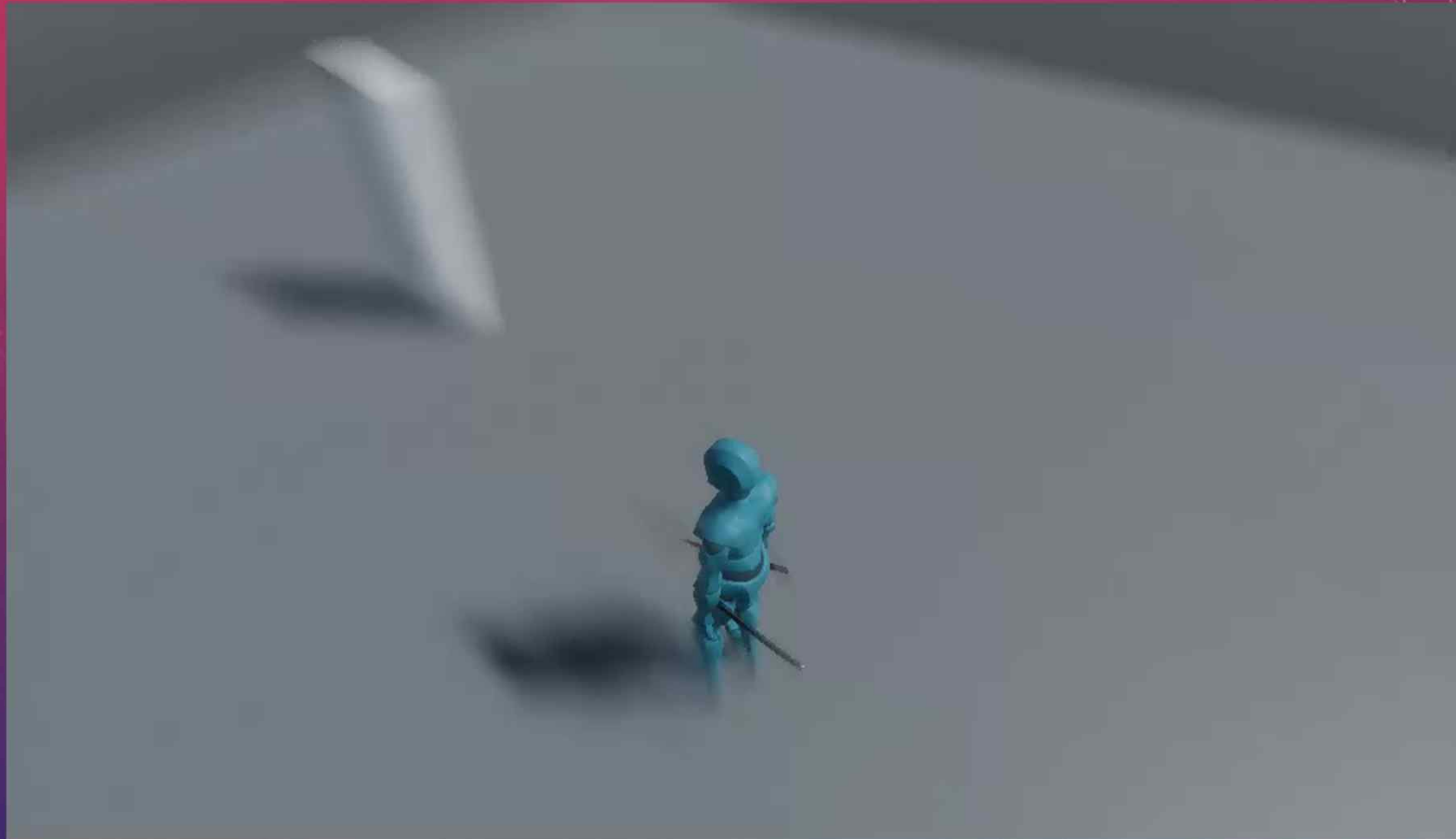
```
    rb.AddExplosionForce(force, go.transform.position, 20); //물체를 자를 때 힘을 가합니다.
```

```
}
```

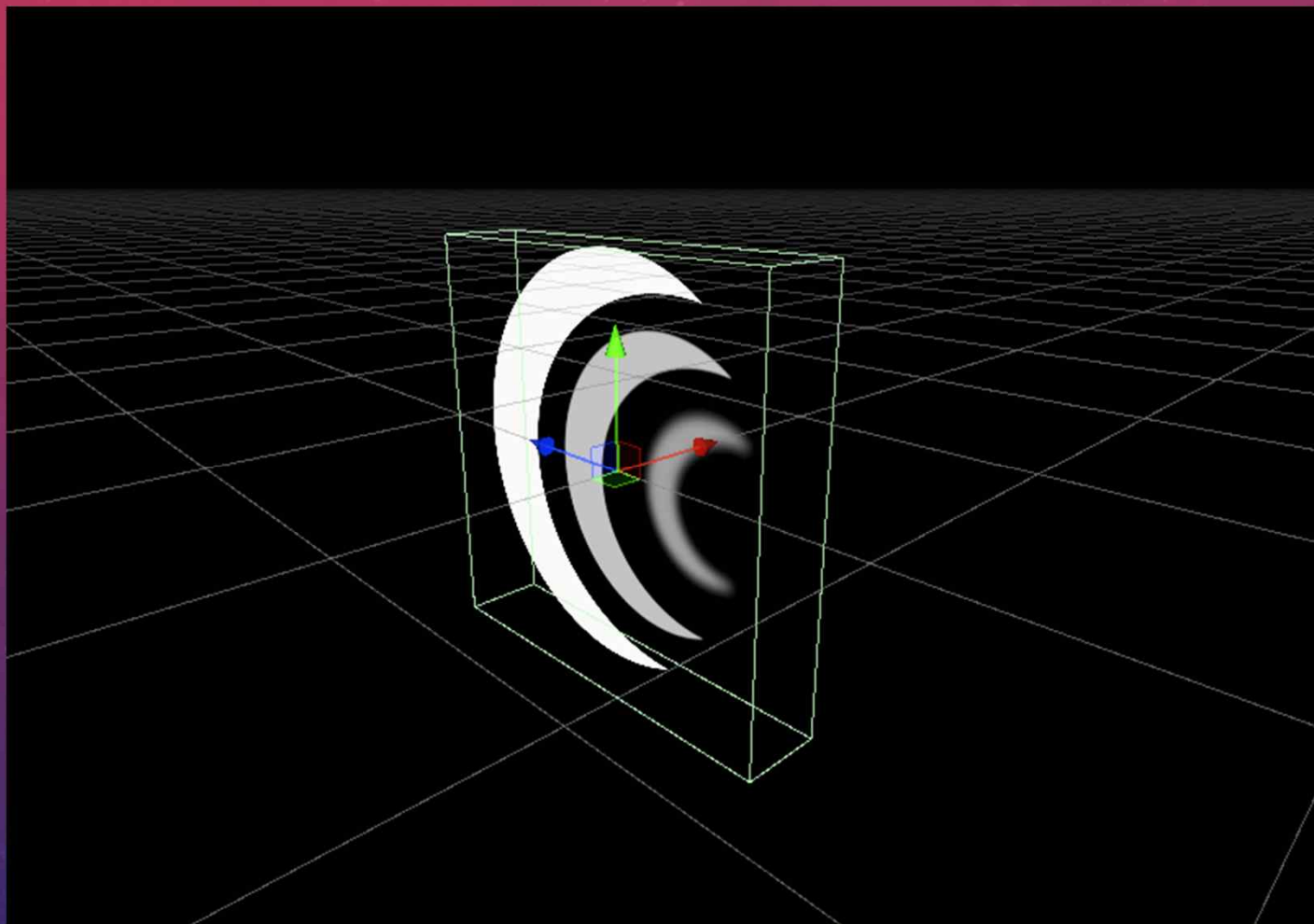
Hull

```
Скрипт Unity | Ссылка: 1
public class Hull : MonoBehaviour
{
    //시간이 지나면 물체를 파괴
    public float lifeTime = 5f;
    Сообщение Unity | Ссылка: 0
    void Start()
    {
        Destroy(gameObject, lifeTime);
    }
}
```

파동 공격

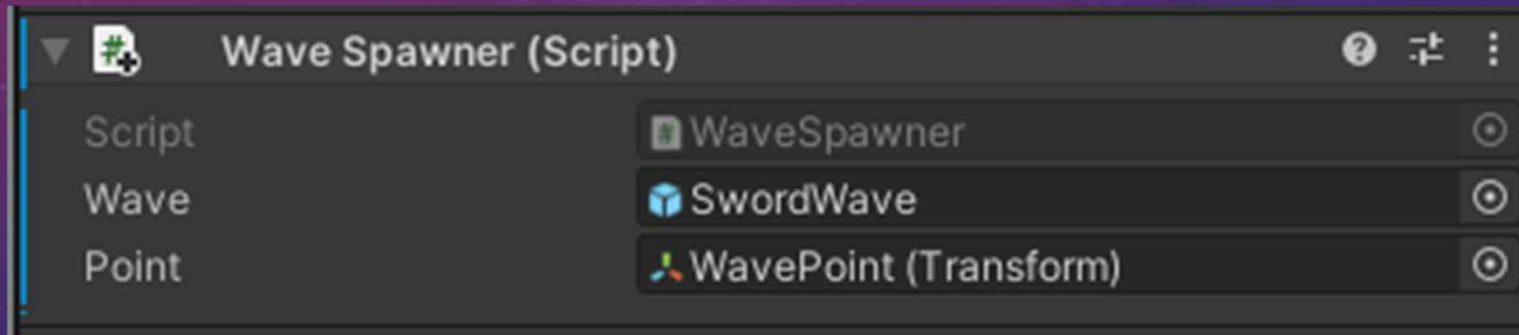


파동



파동 생성기

```
using UnityEngine;
//파동 공격(파동 생성기)
Скрипт Unity (ссылка на ресурсы: 2) | Ссылка: 0
public class WaveSpawner : MonoBehaviour
{
    public GameObject wave; //예제 개체
    public Transform point; //시작 위치
    //특정 각도의 파동 생성(애니메이션에서 호출)
    Ссылка: 0
    public void Spawn(float angle)
    {
        Quaternion newRot = Quaternion.Euler(point.rotation.eulerAngles.x, point.rotation.eulerAngles.y, angle);
        var instance = Instantiate(wave, point.position, newRot);
    }
}
```



파동 스크립트

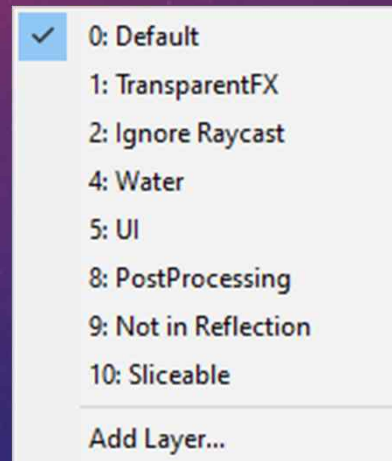
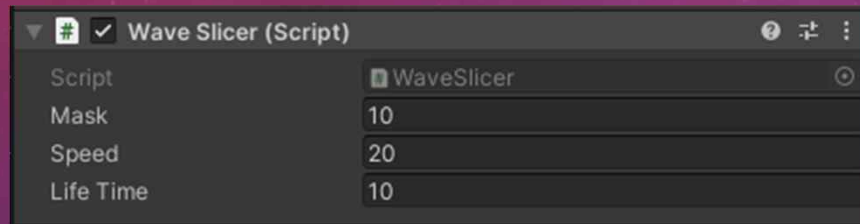
```
using System.Collections.Generic;
using UnityEngine;
//파동
* 스크립트 Unity (1 ссылка на ресурсы) | Ссылка: 0
public class WaveSlicer : MonoBehaviour
{
    List<GameObject> sliced = new List<GameObject>(); //이미 슬라이스된 객체 목록
    public int mask; //LayerMask
    public float speed;
    public float lifeTime;
    Rigidbody rb;
    // Start is called before the first frame update
    * Сообщение Unity | Ссылка: 0
    void Start()
    {
        rb = GetComponent<Rigidbody>();
        Destroy(gameObject, lifeTime);
    }

    // Update is called once per frame
    * Сообщение Unity | Ссылка: 0
    void Update()
    {
        rb.velocity = transform.forward * speed;
    }

    //다른 객체와 충돌할 때 호출됩니다.
    * Сообщение Unity | Ссылка: 0
    private void OnTriggerEnter(Collider other)
    {
        //객체가 슬라이스 가능한지 확인
        if (other.gameObject.layer != mask) return;
        //객체가 이미 슬라이스되었는지 확인
        if (sliced.Contains(other.gameObject)) return;
        (GameObject,GameObject) hulls = ObjectSlicer.Instance.Slice(transform.position,transform.right, other.gameObject);
        sliced.Add(hulls.Item1);
        sliced.Add(hulls.Item2);
    }
}
```

파동 변수

```
List<GameObject> sliced = new List<GameObject>(); //이미 슬라이스된 객체 목록
public int mask; //LayerMask
public float speed;
public float lifeTime;
Rigidbody rb;
```



파동 이동

Сообщение Unity | Ссылок: 0

```
void Update()
```

```
{
```

```
    rb.velocity = transform.forward * speed; //계속 앞으로 이동합니다.
```

```
}
```

파동 히트박스 함수

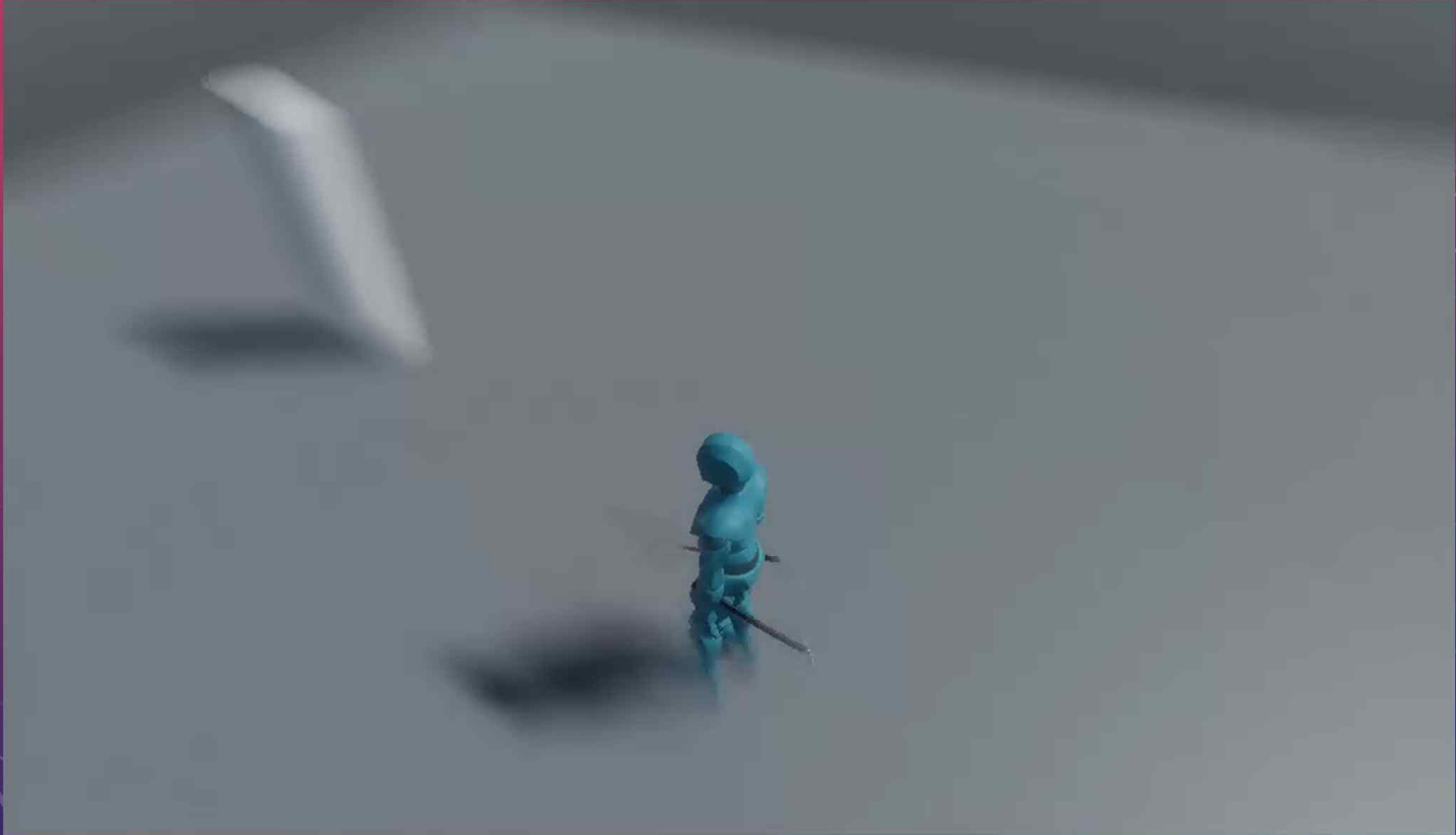
//다른 객체와 충돌할 때 호출됩니다.

Сообщение Unity | Ссылка: 0

```
private void OnTriggerEnter(Collider other)
```

```
{  
    //객체가 슬라이스 가능한지 확인  
    if (other.gameObject.layer != mask) return;  
    //객체가 이미 슬라이스되었는지 확인  
    if (sliced.Contains(other.gameObject)) return;  
    (GameObject,GameObject) hulls = ObjectSlicer.Instance.Slice(transform.position,transform.right, other.gameObject);  
    sliced.Add(hulls.Item1);  
    sliced.Add(hulls.Item2);  
}
```

적



적 생성기



적 스크립트

Скрипт Unity (ссылка на ресурсы: 2) | Ссылка: 2
public class Enemy : MonoBehaviour

```
{  
    public Transform target; //대상(캐릭터)
```

```
    Rigidbody rb;
```

```
    public float speed;
```

```
    // Start is called before the first frame update
```

```
    Сообщение Unity | Ссылка: 0
```

```
    void Start()
```

```
{
```

```
        rb = GetComponent<Rigidbody>();
```

```
}
```

```
    // Update is called once per frame
```

```
    Сообщение Unity | Ссылка: 0
```

```
    void Update()
```

```
{
```

```
    //앞으로 움직
```

```
    Vector3 newVel = transform.forward * speed;
```

```
    newVel.y = rb.velocity.y;
```

```
    rb.velocity = newVel;
```

```
    //대상(캐릭터)를 봅니다.
```

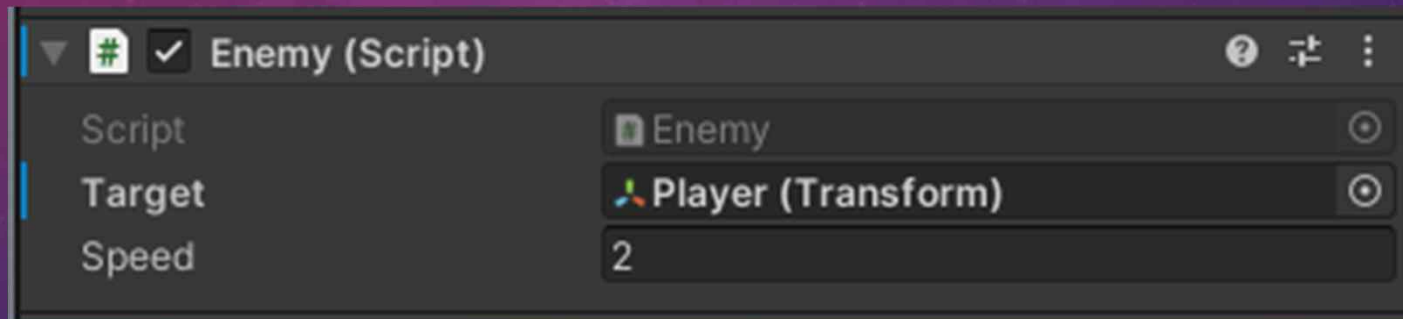
```
    transform.rotation = Quaternion.Euler(0, Quaternion.LookRotation(transform.position - target.position).eulerAngles.y - 180, 0);
```

```
}
```

```
}
```

적 변수

```
public Transform target; //대상(캐릭터)  
Rigidbody rb;  
public float speed;
```



적 이동

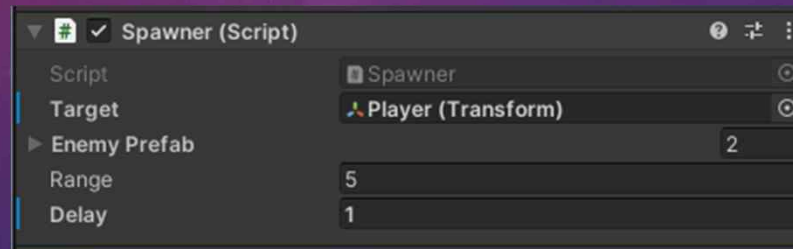
```
// Update is called once per frame
Сообщение Unity | Ссылка: 0
void Update()
{
    //앞으로 움직
    Vector3 newVel = transform.forward * speed;
    newVel.y = rb.velocity.y;
    rb.velocity = newVel;
    //대상(캐릭터)를 봅니다.
    transform.rotation = Quaternion.Euler(0,Quaternion.LookRotation(transform.position - target.position).eulerAngles.y - 180,0);
}
```

적 생성기 스크립트

```
using UnityEngine;
//적 생성기
* Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
public class Spawner : MonoBehaviour
{
    public Transform target; //대상(캐릭터)
    public GameObject[] enemyPrefab; //예제 개체들
    public float range; //생성 범위
    public float delay; //지연
    // Start is called before the first frame update
    * Сообщение Unity | Ссылка: 0
    void Start()
    {
        InvokeRepeating(nameof(Spawn), 0, delay); //지연으로 반복적으로 적을 생성
    }
    //적을 생성
    Ссылка: 1
    void Spawn()
    {
        int randIndex = Random.Range(0, enemyPrefab.Length); //임의의 적
        GameObject obj = Instantiate(enemyPrefab[randIndex], transform.position + transform.right * Random.Range(-range, range), Quaternion.identity); //생성
        Enemy enemy = obj.GetComponent<Enemy>();
        enemy.target = target; //대상 할당
    }
}
```

적 생성기 변수

```
public Transform target; //대상(캐릭터)  
public GameObject[] enemyPrefab; //예제 개체들  
public float range; //생성 범위  
public float delay; //지연
```



적 생성 함수

```
// Start is called before the first frame update
Сообщение Unity | Ссылка: 0
void Start()
{
    InvokeRepeating(nameof(Spawn), 0, delay); //지연으로 반복적으로 적을 생성
}
//적을 생성
Ссылка: 1
void Spawn()
{
    int randIndex = Random.Range(0, enemyPrefab.Length); //임의의 적
    GameObject obj = Instantiate(enemyPrefab[randIndex], transform.position + transform.right * Random.Range(-range, range), Quaternion.identity); //생성
    Enemy enemy = obj.GetComponent<Enemy>();
    enemy.target = target; //대상 할당
}
```

끝