

**Yapay Zeka ile
Oyun geliştirme**

Oyunu Nasıl Eğlenceli Hale Getiririz?

Neye çok
ihtiyacımız yok?

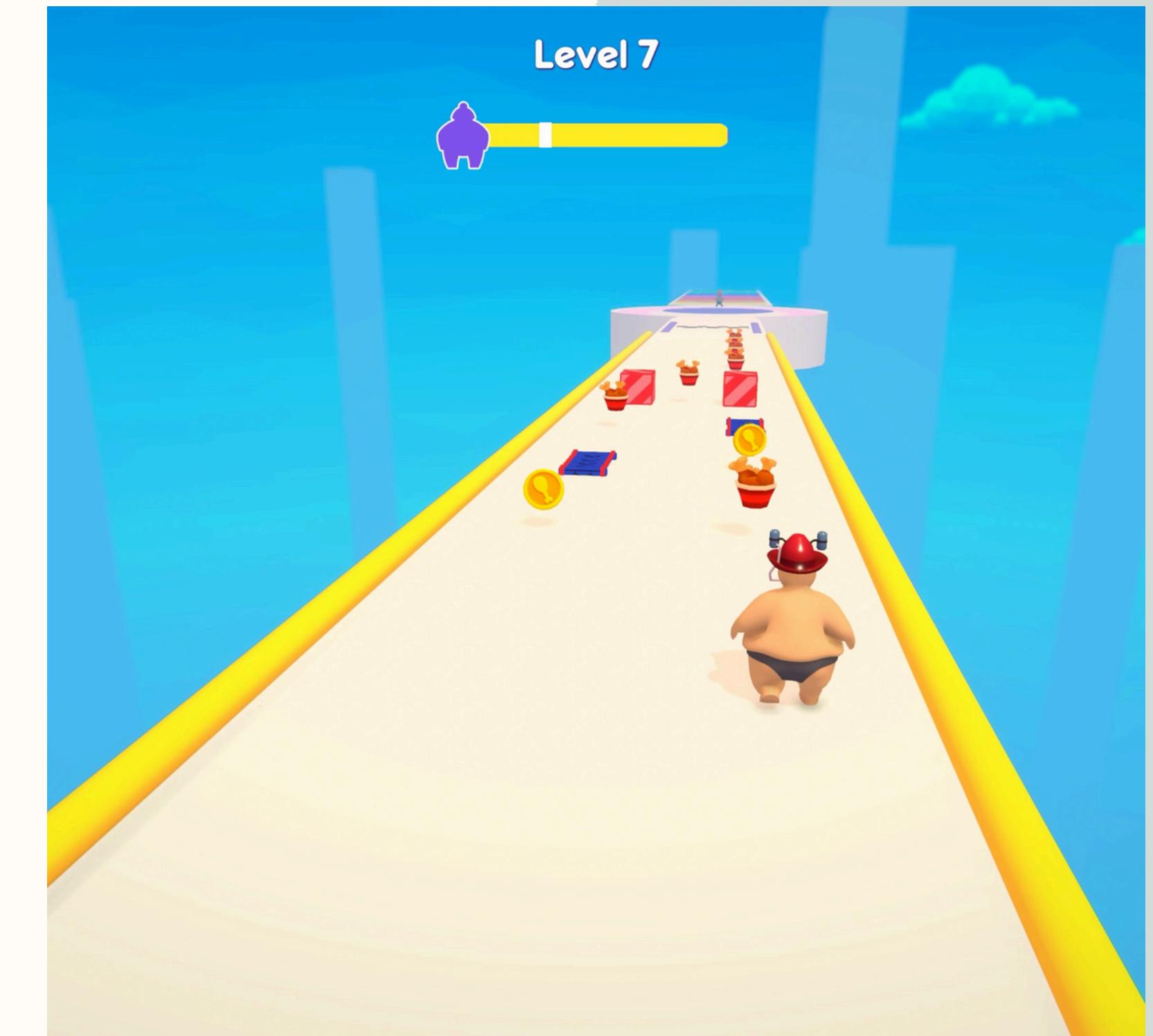
- Mekanikler
- Hissiyat
- Mücadele
- Soslar(Hikaye vb.)
- Kapsamlı Dokuman
- Uzun süren prototip süreci

Plansız AI

Yapay zekayı plansız bir şekilde kullandığımızda çok jenerik sonuçlar almamız kaçınılmaz oluyor bu nedenle ilk önce ne istediğimizi bilerek işe başlamalıyız.

Başlangıçta oyunun tamamını düşünmek yerine belirli konseptleri ve aklındaki basit fikirleri birleştirmek yeterli olacaktır.

Amatörler olarak amacımız muazzam bir oyun yapmak değil prototipleyip eğlenceli olup olmadığını kontrol etmek olmalı.





Mekanikler

Temel atmak

Temelleri oluşturacak konseptler, önceden düşünülebilir veya yapay zeka ile danışarak edinilebilir

Bir oyun yapmak istiyorum içerisinde ne olursa eğlenceli olur?

Güzel soru 

Bir oyunu eğlenceli yapan şey, grafiklerden çok oyuncuya ne hissettiğidir. Aşağıda hem genel prensipleri hem de somut oyun fikirlerini vereyim.

⌚ Eğlencenin temel formülü

Bir oyunda en az 2–3 tanesi mutlaka olmalı:

1. Seçim yapma (oyuncu karar versin)
2. Gelişim hissi (güçleniyorum duygusu)
3. Risk–ödül (risk alırsam daha çok kazanırım)
4. Sürpriz (her seferinde aynı olmasın)
5. Kısa ama tatmin edici döngü (5–10 dakikada “bir tur daha” dedirtsin)

İçinde bir dev tarafından kovalandığımız ve bu devden kancalı bir silahla kaçınırken arada ateş ederek zarar verdigimiz türden bir oyun yapmak istiyorum bana aşama aşama yardım et ve sahneyi kuralım.

Harika bir fikir 

Dev tarafından kovalanma + kancalı silahla kaçış + ateş ederek zarar verme = çok güçlü bir core gameplay.

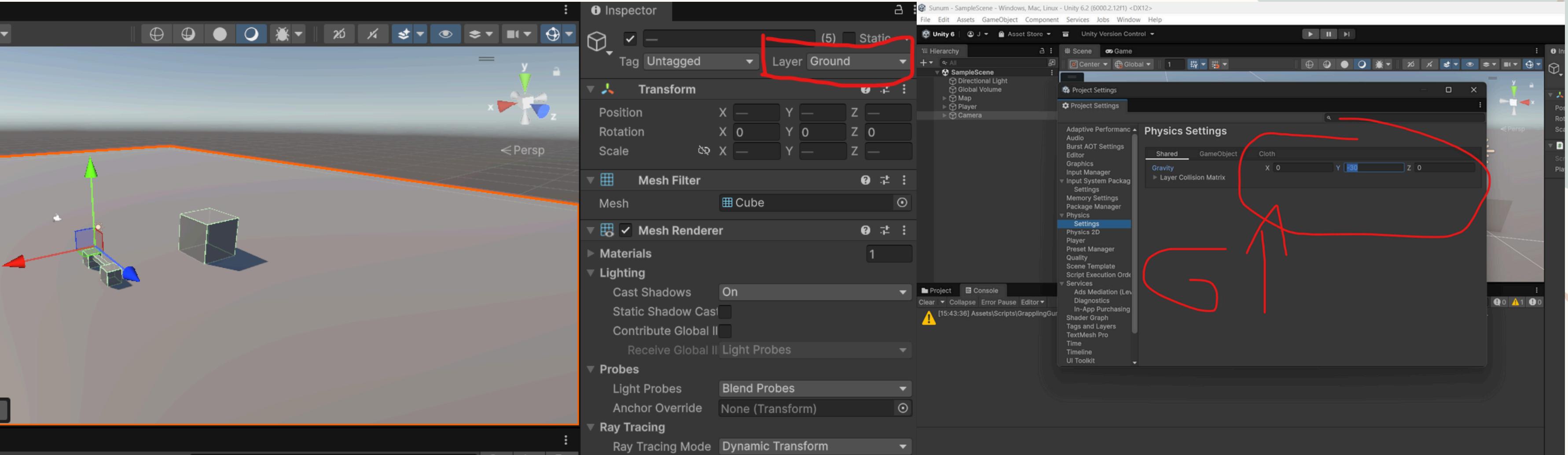
Bunu Unity'de adım adım, gerçekten sahneyi kurarak yapalım.

Acele etmeyeceğiz, her aşama oynanabilir olacak.

Aşağıda YOL HARİTASI var, sonra AŞAMA 1'i birlikte yapıyoruz 

Ben 2. görseldeki önceden kurguladığım yolu seçeceğim.

Mekanikleri en ilkel haliyle oluşturuyoruz-> Karakter hareketi, Kanca atma



kanca atılacak yerleri belirledik

```
using System;
using UnityEngine;

0 references
public class PlayerMovement : MonoBehaviour {

    2 references
    public Transform playerCam;
    7 references
    public Transform orientation;

    26 references
    private Rigidbody rb;
    4 references
```

Karakter hareketi

Yerçekimini artttırdık.

```
using UnityEngine;
1 reference
public class GrapplingGun : MonoBehaviour {
    5 references
    private LineRenderer lr;
    5 references
    private Vector3 grapplePoint;
    1 reference
    public LayerMask whatIsGrapable;
    2 references | 2 references | 2 references
    public Transform gunTip, camera, player;
    1 reference
    private float maxDistance = 100f;
    11 references
    private SpringJoint joint;
```

Kanca kodu

Mekanikleri oluşturuyoruz-> Düşman takip, Ateş etme

Projenize Eklemeniz Gereken Kod (EnemyAI.cs):

```
C#  
  
using UnityEngine;  
using UnityEngine.AI;  
  
public class EnemyAI : MonoBehaviour {  
    public Transform player;  
    public NavMeshAgent agent;  
    public LayerMask whatIsGround, whatIsPlayer;  
  
    // Titan Stats  
    public float health = 100f;  
  
    private void Awake() {  
        agent = GetComponent<NavMeshAgent>();  
    }  
  
    private void Update() {  
        if (player != null) {  
            // Oyuncuya kovala  
            agent.SetDestination(player.position);  
        }  
    }  
  
    public void TakeDamage(float damage) {  
        health -= damage;  
  
        // Vurulma efekti veya rengi değişebilir (Sunum için opsiyonel görsel şo  
GetComponent<Renderer>().material.color = Color.red;  
Invoke("ResetColor", 0.1f);  
  
        if (health <= 0) Die();  
    }  
}
```

Düşman takibi

Projenize Eklemeniz Gereken Kod (GunSystem.cs):

```
C#  
  
using UnityEngine;  
  
public class GunSystem : MonoBehaviour {  
    public float damage = 10f;  
    public float range = 100f;  
    public Camera fpsCam;  
    public ParticleSystem muzzleFlash; // Ateş efekti (varsa)  
  
    void Update() {  
        // Sağ tık ile ateş et  
        if (Input.GetMouseButtonDown(1)) {  
            Shoot();  
        }  
    }  
  
    void Shoot() {  
        if (muzzleFlash != null) muzzleFlash.Play();  
  
        RaycastHit hit;  
        // Ekranın tam ortasından işin yolla  
        if (Physics.Raycast(fpsCam.transform.position, fpsCam.transform.forward)) {  
            Debug.Log(hit.transform.name + " vuruldu!");  
  
            EnemyAI target = hit.transform.GetComponent<EnemyAI>();  
            if (target != null) {  
                target.TakeDamage(damage);  
            }  
  
            // Vuruş efekti eklenebilir (Rigidbody'ye güç uygula)  
            if (hit.rigidbody != null) {  
                hit.rigidbody.AddForce(-hit.normal * 30f);  
            }  
        }  
    }  
}
```

Ateş Etme kodları

Adım 3: Unity Editör İçindeki Kurulum (Çok Önemli!)

Kodları yapıştırdıktan sonra sunumda "Demo" yaparken hata almamak için şu ayarları yapmalısınız:

1. NavMesh (Zemin):

- Zemin objenize (Plane/Terrain) tıklayın.
- Sağ üstten **Static** kutucuğunu işaretleyin.
- Window > AI > Navigation sekmensinden **Bake** butonuna basın. (Mavi bir harita olmalıdır).

2. Titan (Düşman):

- Sahneye bir **Capsule** veya **Cube** oluşturun.
- Scale değerlerini X:3, Y:8, Z:3 yapın (Dev gibi görünsun).
- Bu objeye **NavMeshAgent** componenti ekleyin.
- Scripti (`EnemyAI`) ekleyin ve "Player" slotuna oyuncunuzu sürükleyin.
- **ÇOK ÖNEMLİ:** Objeye Inspector'in en üstünden **Tag** kısmından "Add Tag" diyip "**Enemy**" tag'i oluşturun ve bu tag'i seçin. Yoksa silah tanımaz.

3. Silah:

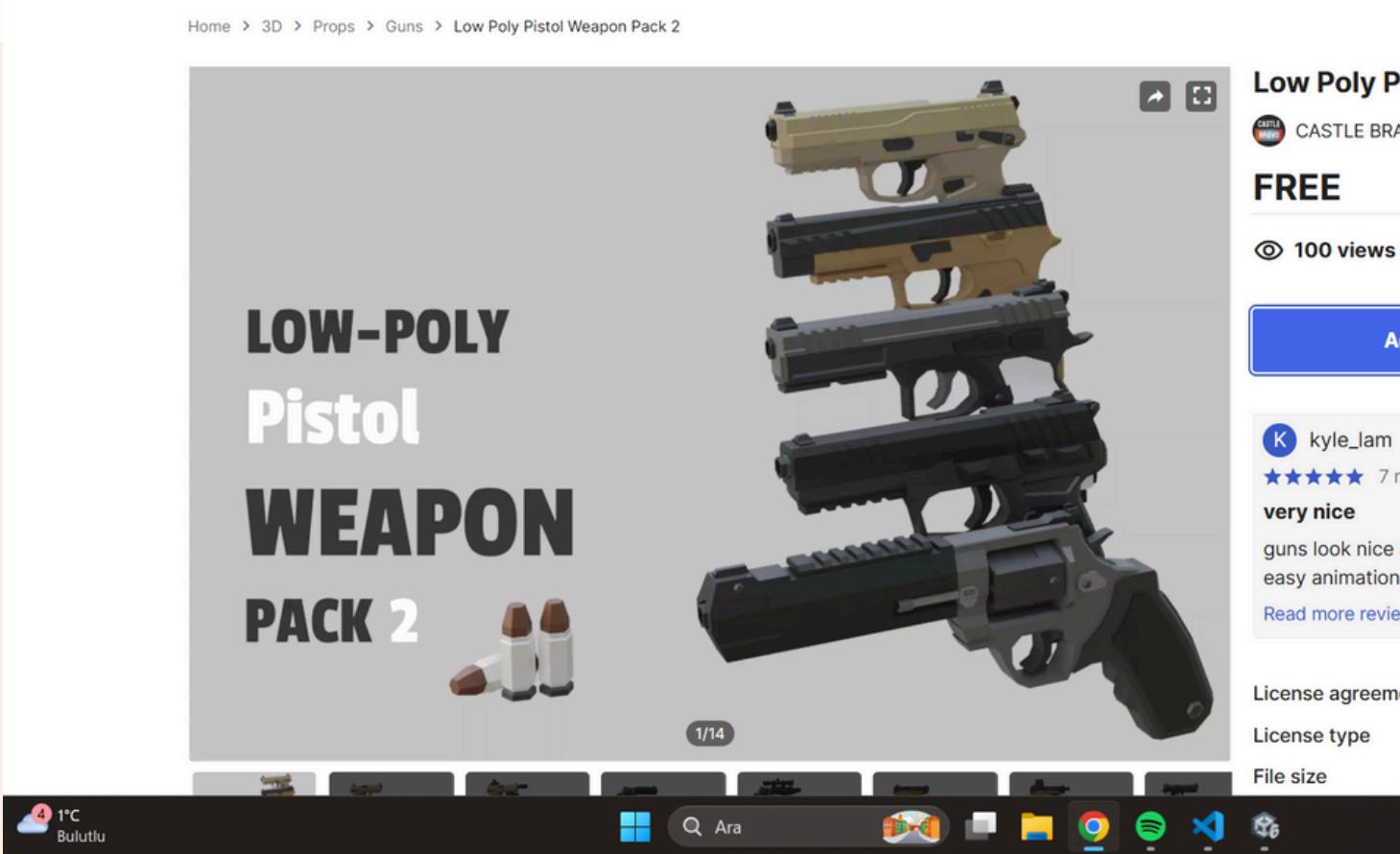
- Silah objenize (Kamera altındaki) **GunSystem** scriptini atın.
- FPS Cam kısmına kameranızı sürükleyin.

Sahne'ye eklemek için yapmamız gerekenler

Hissiyat



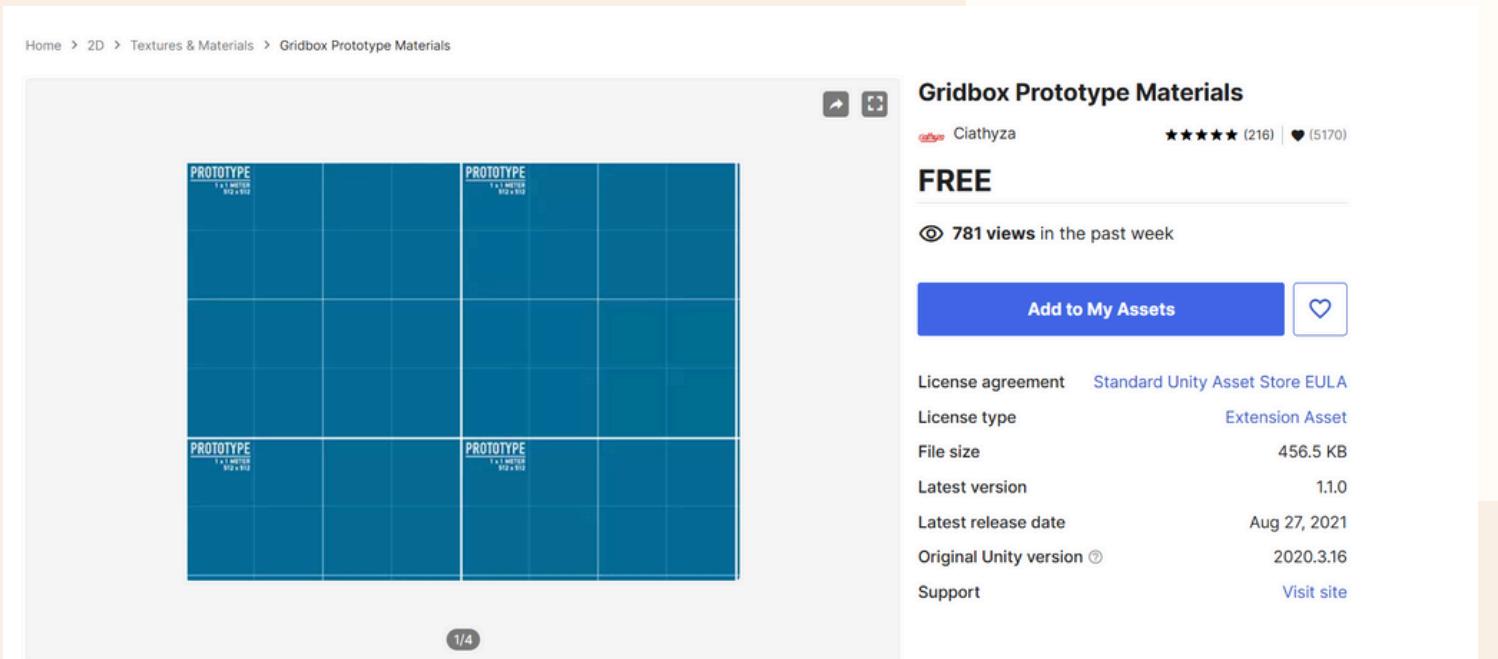
Var olan mekanikleri daha iyi hale getirmeye çalışıyoruz



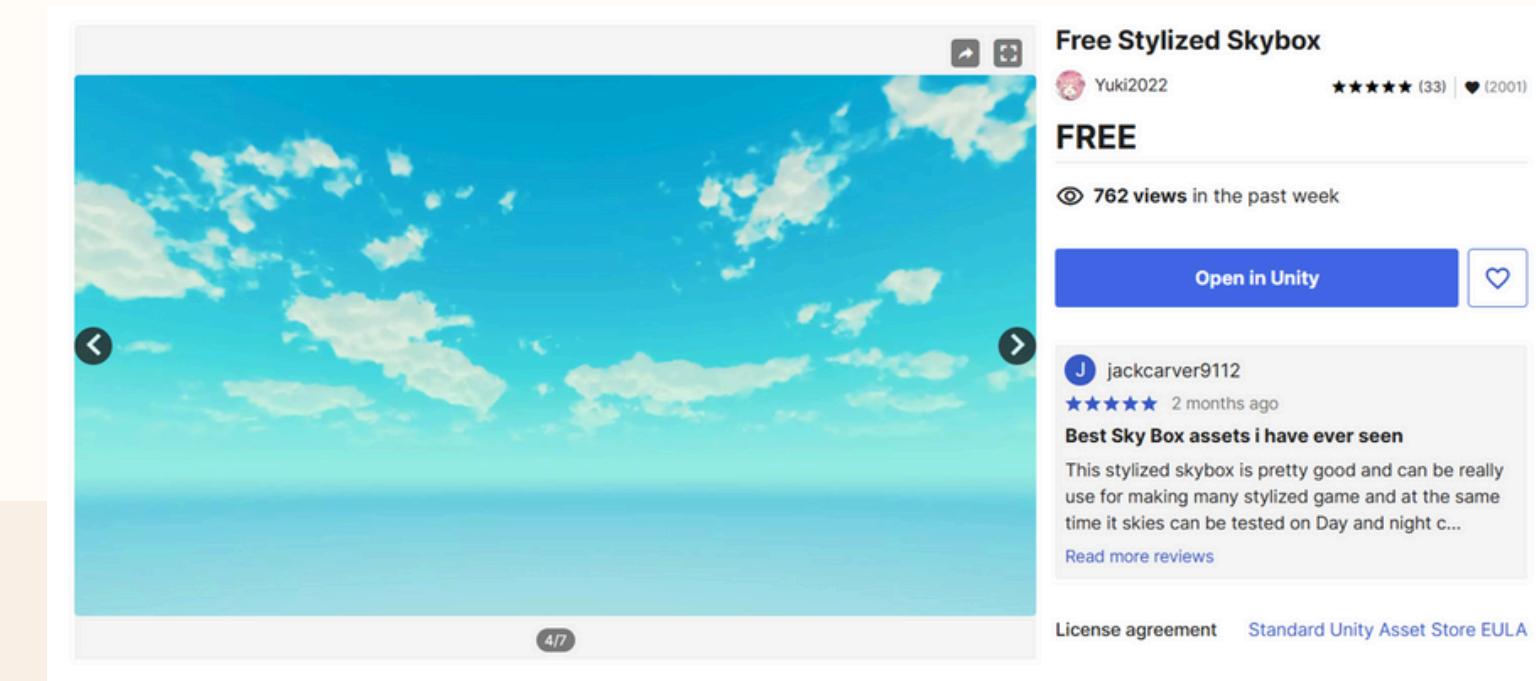
Silah assetimiz



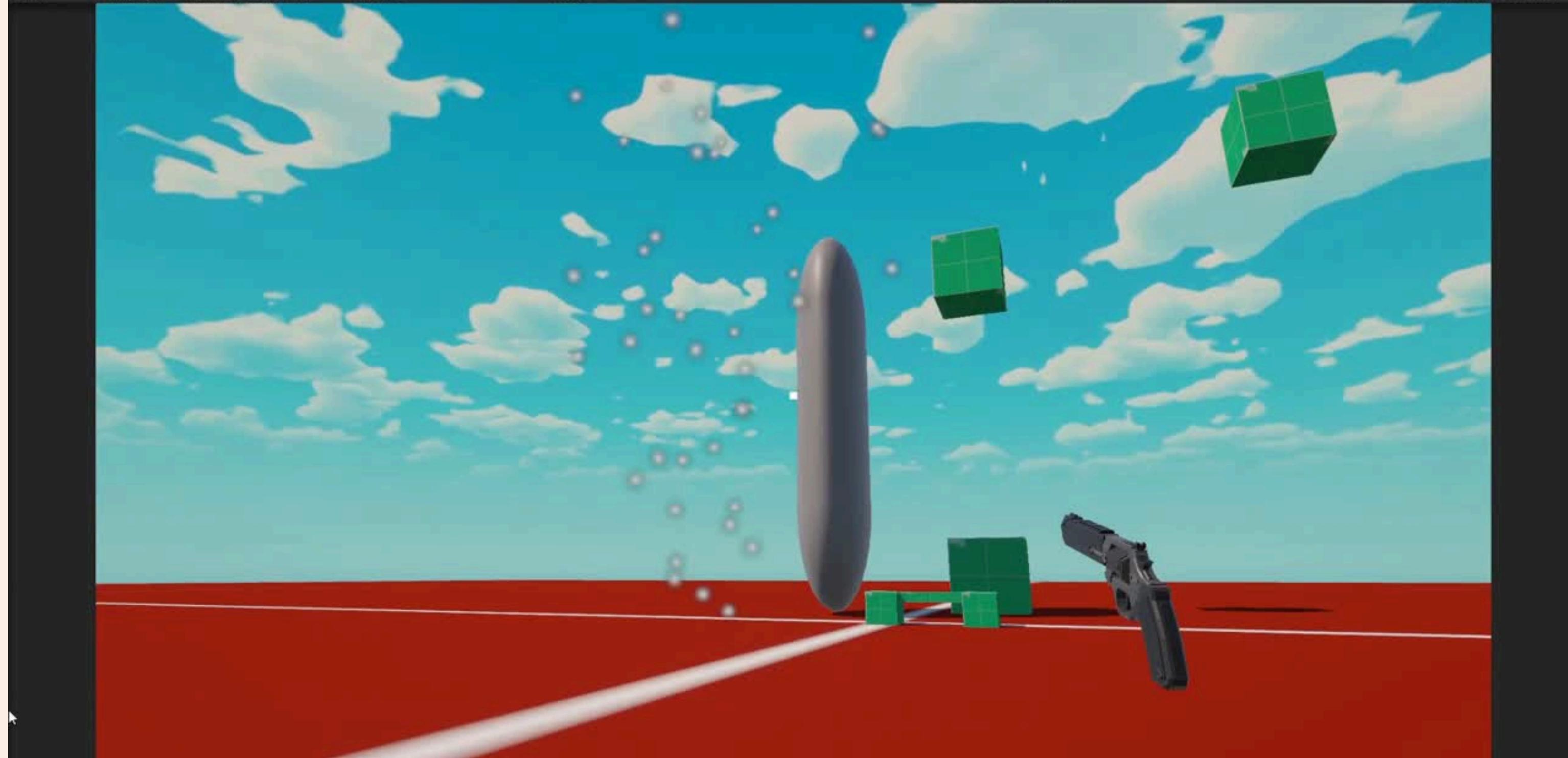
Kanca Görünüm



Prototipe uygun material



Gökyüzü görseli



Bariz olarak kötü hissettirenler-> ateş etme, düşman

Bir tespit yaptık ve bunun için hissiyatı iyileştiren isteklerimizi AI'a yazdık

1. Adım: Kamera Titremesi (Screen Shake)

Önce kamerayı sarsacak scripti yazalım. Bu script, herhangi bir yerden (ateş etme, patlama) çağrılabılır olacak.

Script Adı: CameraShake.cs (Bunu Main Camera objesine ekleyin)

```
c#  
  
using UnityEngine;  
using System.Collections;  
  
public class CameraShake : MonoBehaviour  
{  
    public IEnumerator Shake(float duration, float magnitude)  
    {  
        Vector3 originalPos = transform.localPosition;  
        float elapsed = 0.0f;  
  
        while (elapsed < duration)  
        {  
            float x = Random.Range(-1f, 1f) * magnitude;  
            float y = Random.Range(-1f, 1f) * magnitude;  
  
            transform.localPosition = new Vector3(x, y, originalPos.z);  
  
            elapsed += Time.deltaTime;  
            yield return null;  
        }  
  
        transform.localPosition = originalPos;  
    }  
}
```

silah için bir bullet prefabı ile onun kopyasının olusmasını(mermi gönderip), eğer yere çarpara bulletexplosion olusmasını ve her silahı sıktığımızda muzzle efekt ile, recoil(geri tepme efektinin olmasını) bunun yanında screen shake olmasını istiyorum bunları nasıl yapacağımı detaylıca anlat ve yapalım

Süper 🌟 Bu istedigin şeyler FPS / TPS hissini veren temel silah sistemi ve hepsini temiz, modüler şekilde kurabiliriz.

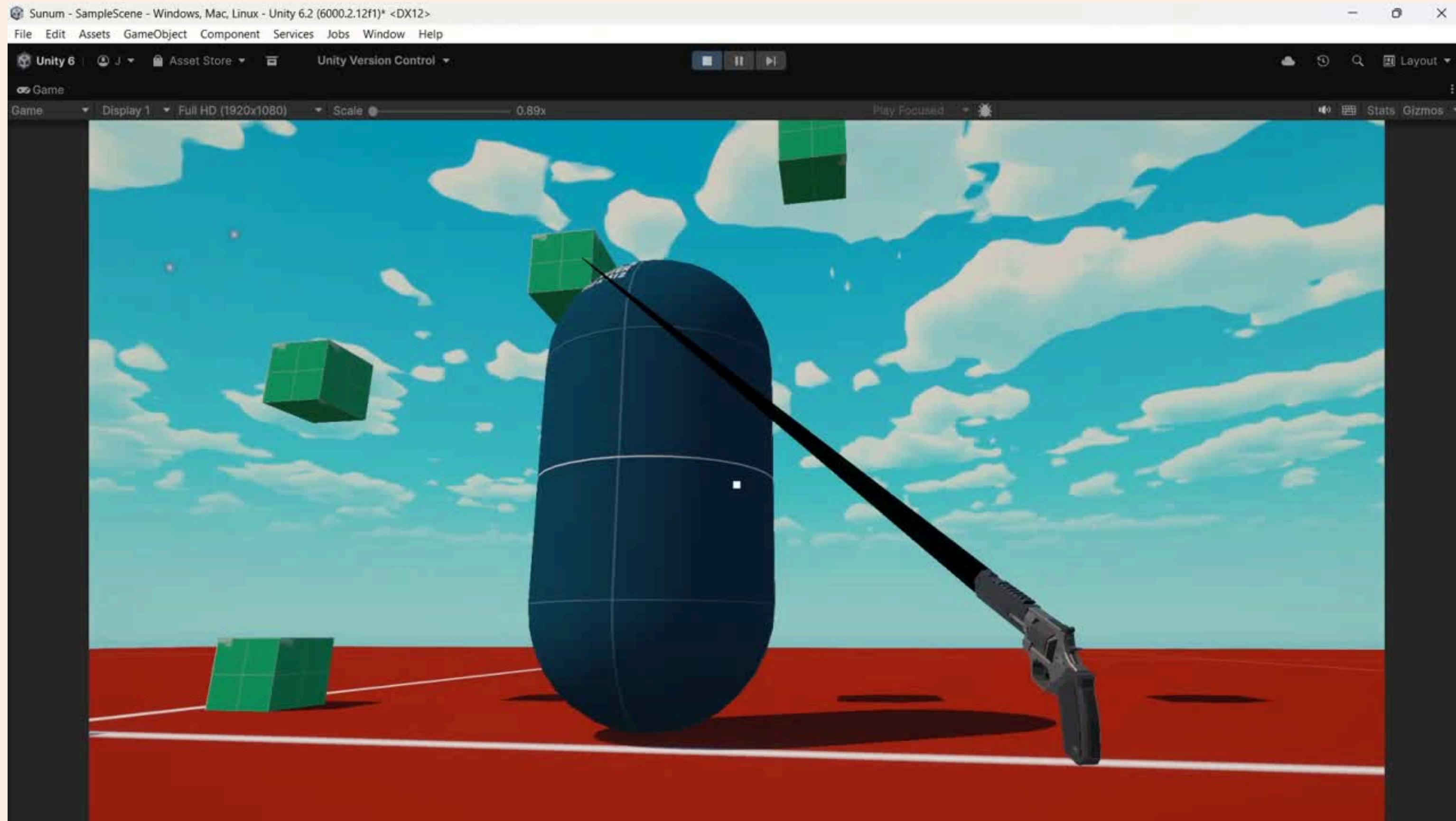
Aşağıda gerçekten sahnede kurarak gideceğiz. Kopyala–yapıştır değil, nedenini anlayarak yapacaksın.

2. Adım: Fizik Tabanlı Mermi (Bullet Prefab)

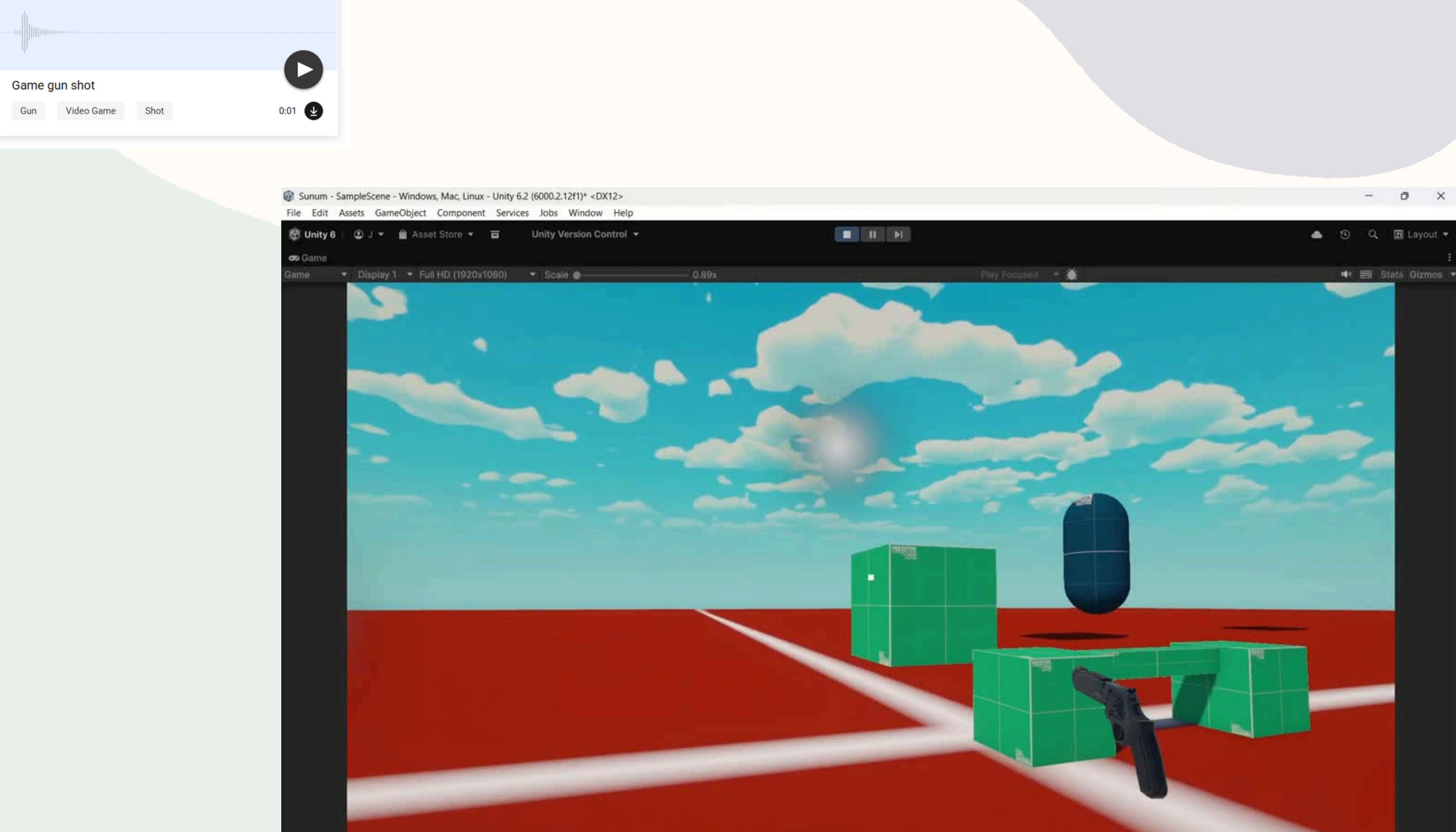
Raycast (ışın) yerine fiziksel bir mermi atacağız. Bu mermi bir şeye çarپınca patlayacak.

Script Adı: BulletController.cs

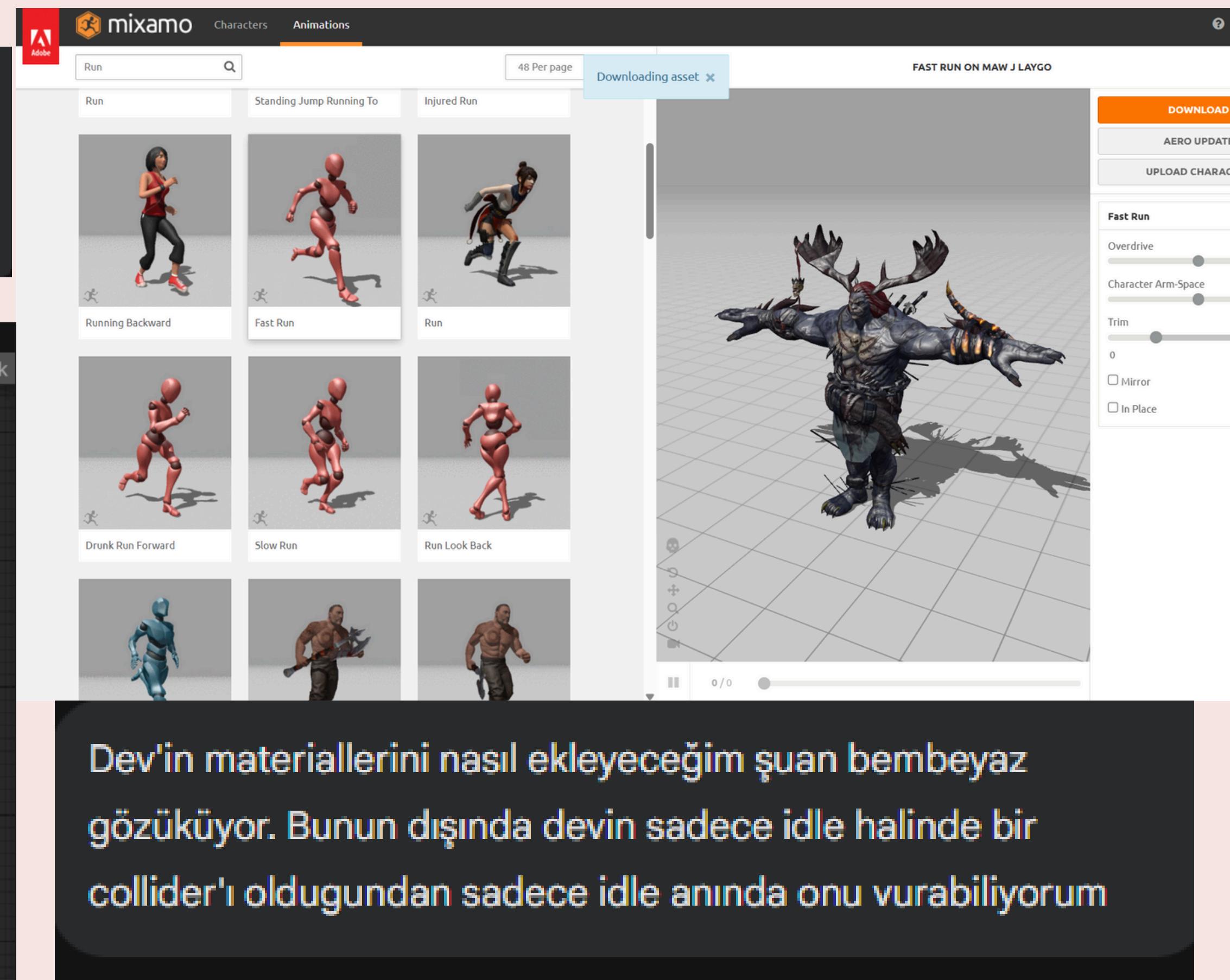
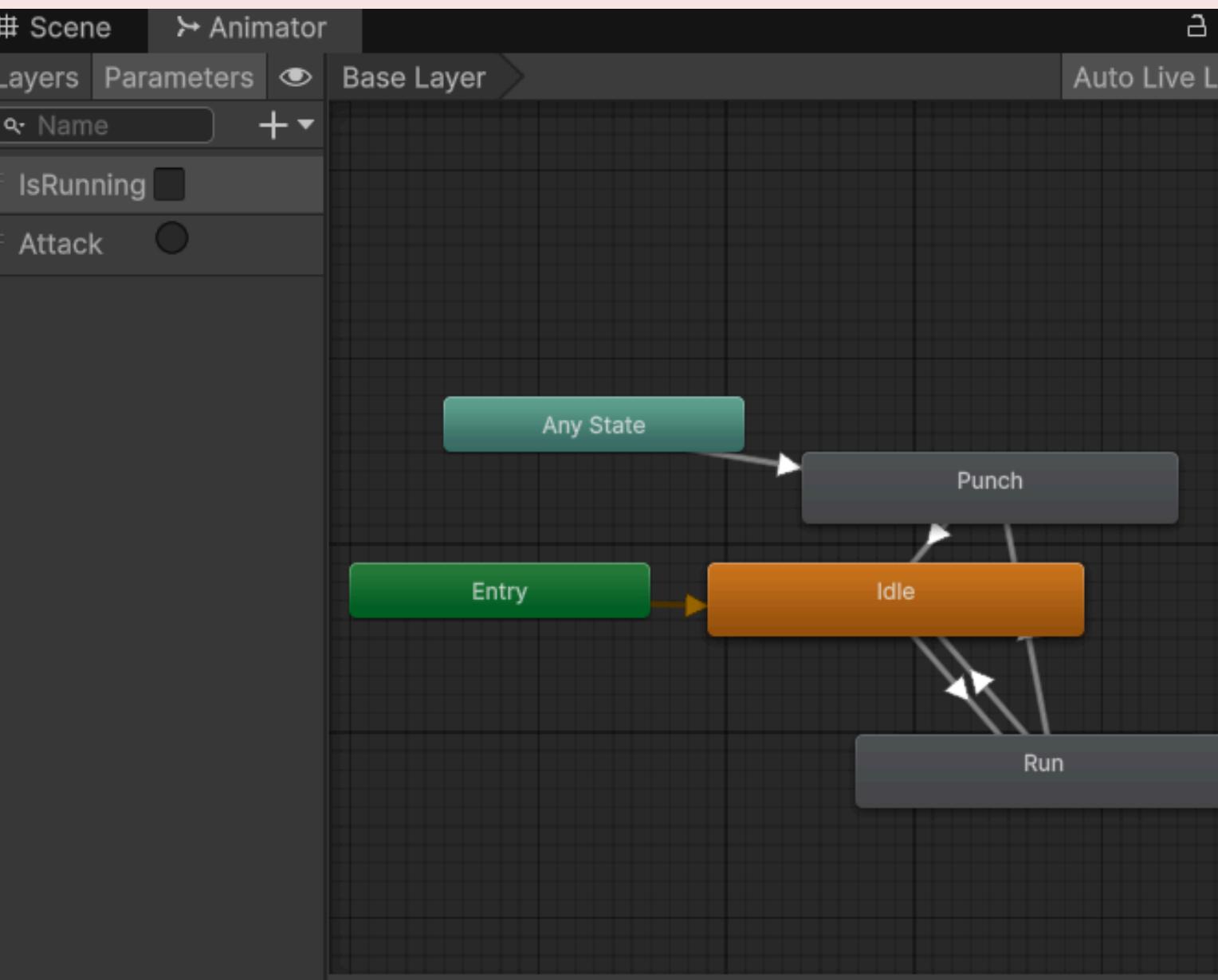
Hissiyatı artırmak için silah teperken y posizyonunu da değiştirip hareketi daha eğlenceli yapmak istiyorum. Bunun dışında mermi ilerlerken, arkasından bir line renderer ilerlemesini istiyorum böylece giden mermiyi daha net görürüz. Bir de silahı ateşlediğimizde ateş etme sesi gelmesini istiyorum.

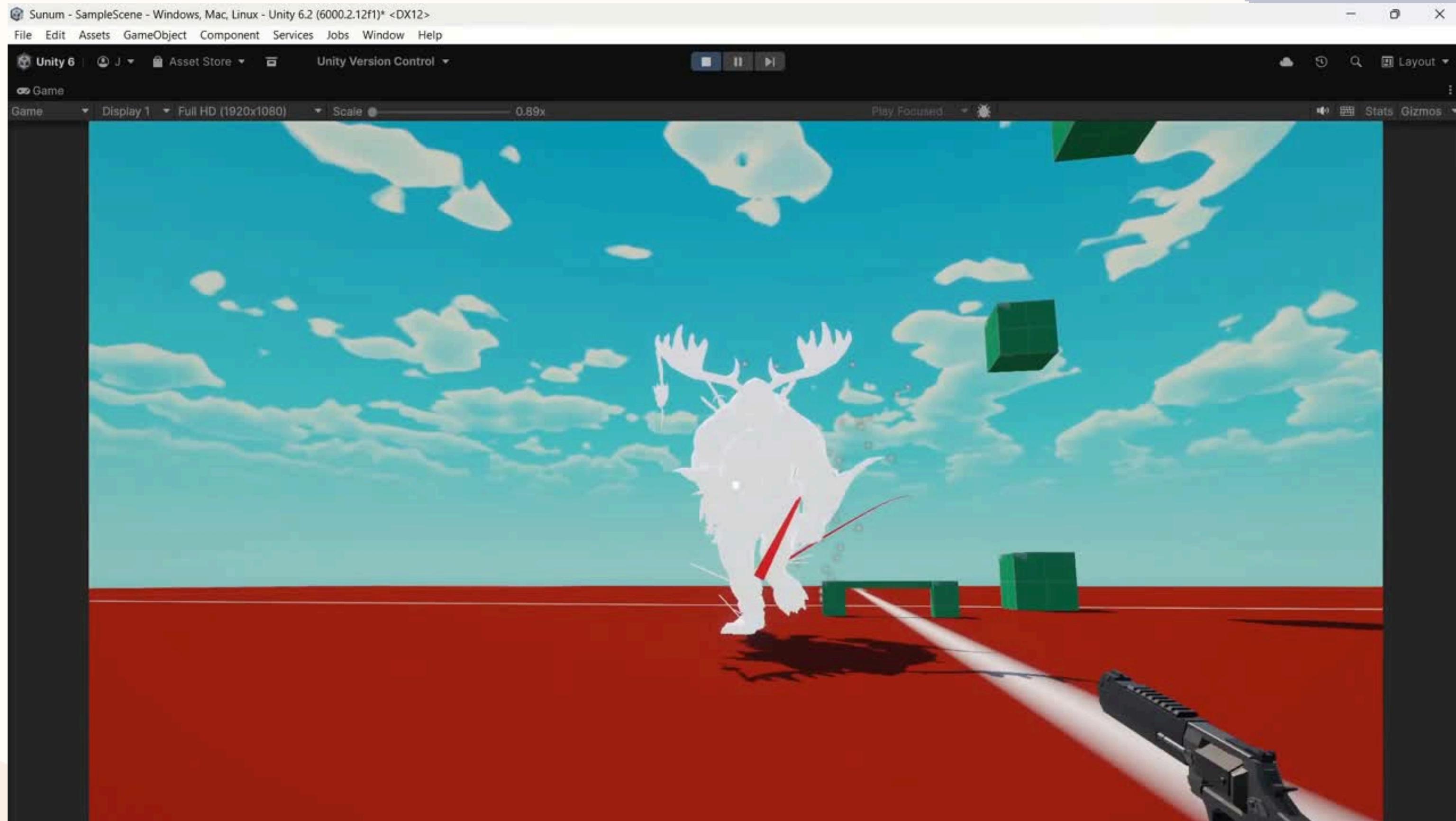


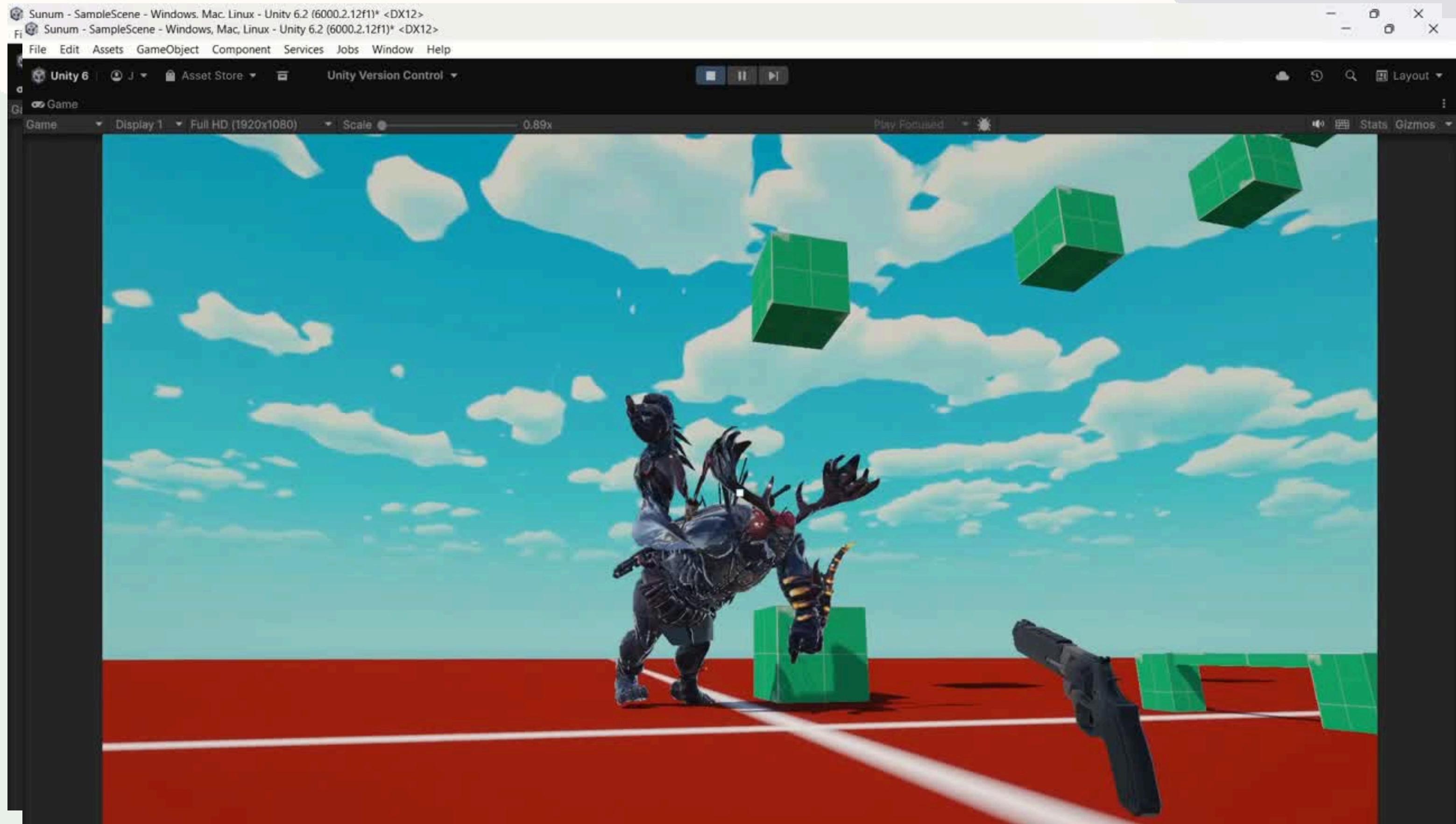
Eksiklik-> Ses



Şimdi bu dev'in bir modeli olmasını ve bize saldırımı yapmasını istiyorum bunu mixamo'dan elde edip yanımıza yaklaştığımızda saldırısını istiyorum bunu yapalım daha sonra diğer mekaniklere gireriz







Dev hasar yiince yediği hasar devin üzerinde gözüksün istiyorum bunu yapalım

Düşünme sürecini göster ▾

Harika bir fikir! Vuruş hissiyatını (impact) tepe noktaya çıkaracak bir detay. RPG ve shooter oyunlarının olmazsa olmazı "Floating Combat Text" (Yüzen Savaş Metni) sistemini ekleyelim.

Bu işlem için 3 aşamalı bir yol izleyeceğiz:

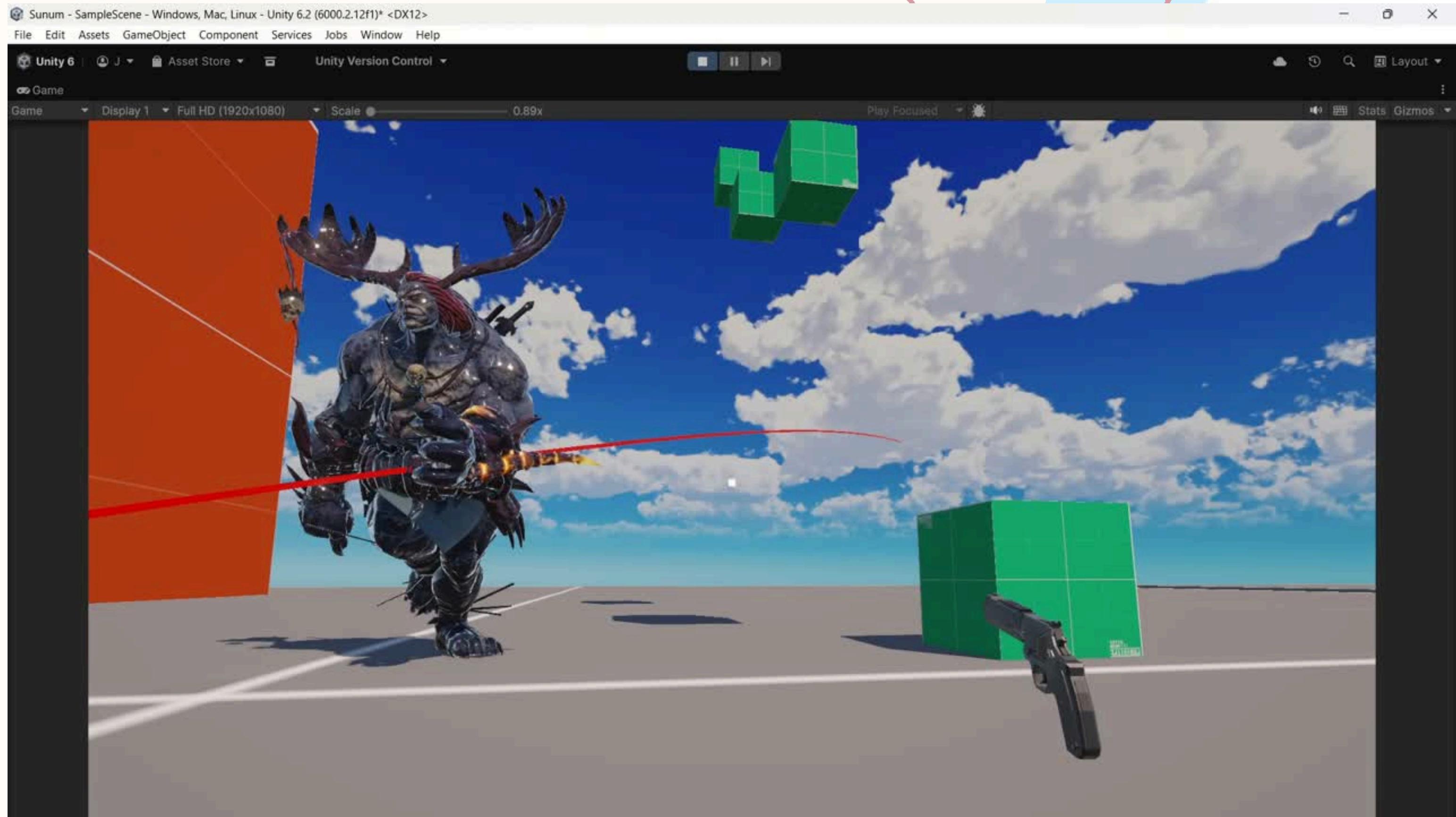
- Prefab Hazırlığı:** Havada süzülecek metnin ayarlanması.
- Popup Scripti:** Metnin yukarı kayıp kaybolmasını sağlayan kod.
- Entegrasyon:** `EnemyAI` scriptinin bu metni ne zaman çıkaracağını bilmesi.

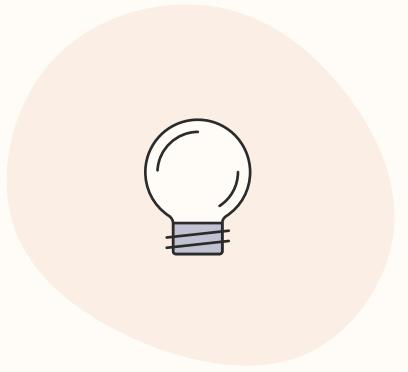
Karakterimize bir walljump özelliği koyalım. Bundan sonra özellik olarak şunu sağlamak istiyorum, dev çok tehlikeli onu sadece kaçip vurarak yenebiliriz bu yüzden başka ne özellikler eklemeliyim sen karar ver.

Projeye Eklemeniz Gereken Kod (`WallRun.cs`):

C#

```
using UnityEngine;  
  
public class WallRun : MonoBehaviour  
{  
    [Header("Movement")]  
    public Transform orientation;  
    public Transform playerCam;  
    private Rigidbody rb;  
  
    [Header("Detection")]  
    public LayerMask whatIsWall;  
    public float wallRunForce = 200f;  
    public float wallJumpUpForce = 7f;  
    public float wallJumpSideForce = 12f;  
    public float maxWallRunTime = 1.5f;  
    private float wallRunTimer;
```





Şimdi yavaş yavaş oyunu bitirmek istiyorum bu oyunun bir bölümü nasıl olmalı sence

◆ Düşünme sürecini göster ▾

Harika bir noktaya geldik. Yarınki sunum için **tam bitmiş bir oyun değil, mekaniklerin parladığı bir "Vertical Slice"** (Oyunun en can alıcı kesiti) hazırlamalıyız.

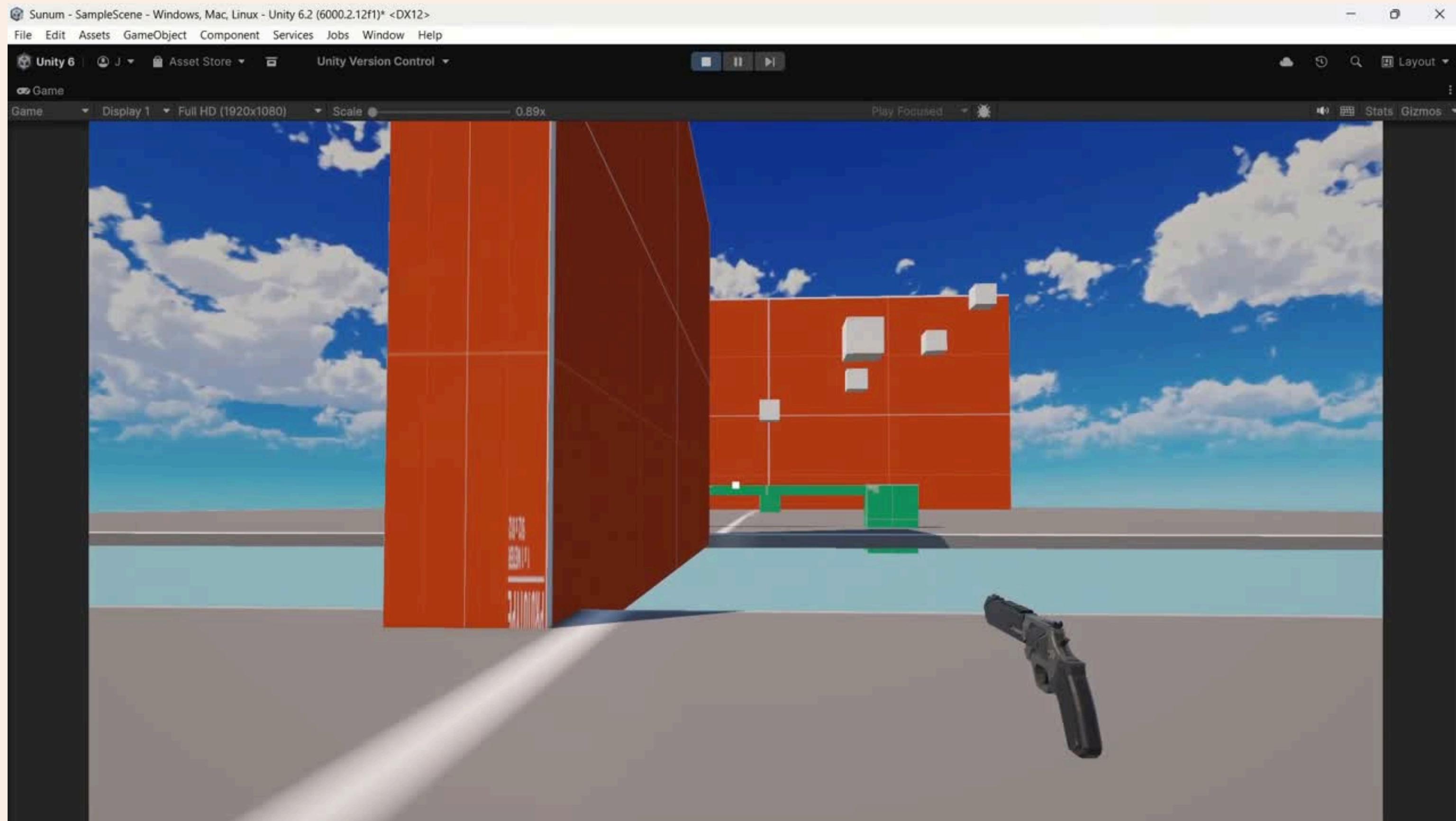
□ Oyunda grabbling gun kullanarak hızlanırken etrafımızdan rüzgar gibi efektler çıkışmasını istiyorum bunu da yapalım

◆ Düşünme sürecini göster ▾

Harika bir fikir! "Hız hissi" (Sense of Speed) bir hareket oyununun olmazsa olmazıdır. Dani'nin oyunlarında da bu "ruzgar çizgileri" (speed lines) efekti sıkça kullanılır.

Bunu yapmak için bir **Particle System** (Parçacık Sistemi) oluşturacağız ve karakterin hızına göre bu sistemin yaydığı parçacık sayısını artırıp azaltacağız.

Hazırsan, Unity'nin görsel efekt mutfağına girelim.



1. Win Panel (Kazanma Ekranı) İçin Prompt

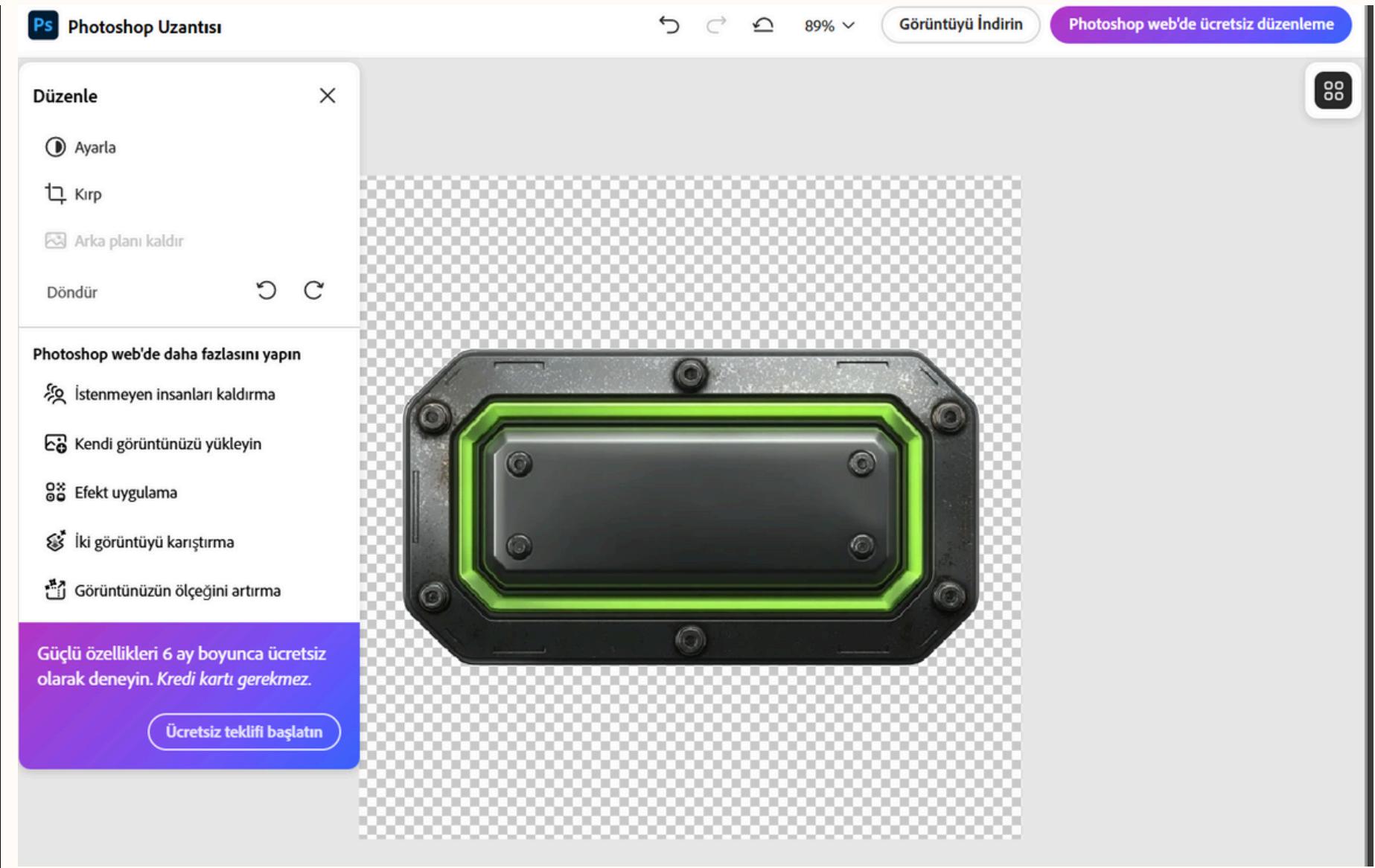
Bu prompt, arkası hafif şeffaf, kenarları teknolojik ve parlak bir pencere oluşturacak.

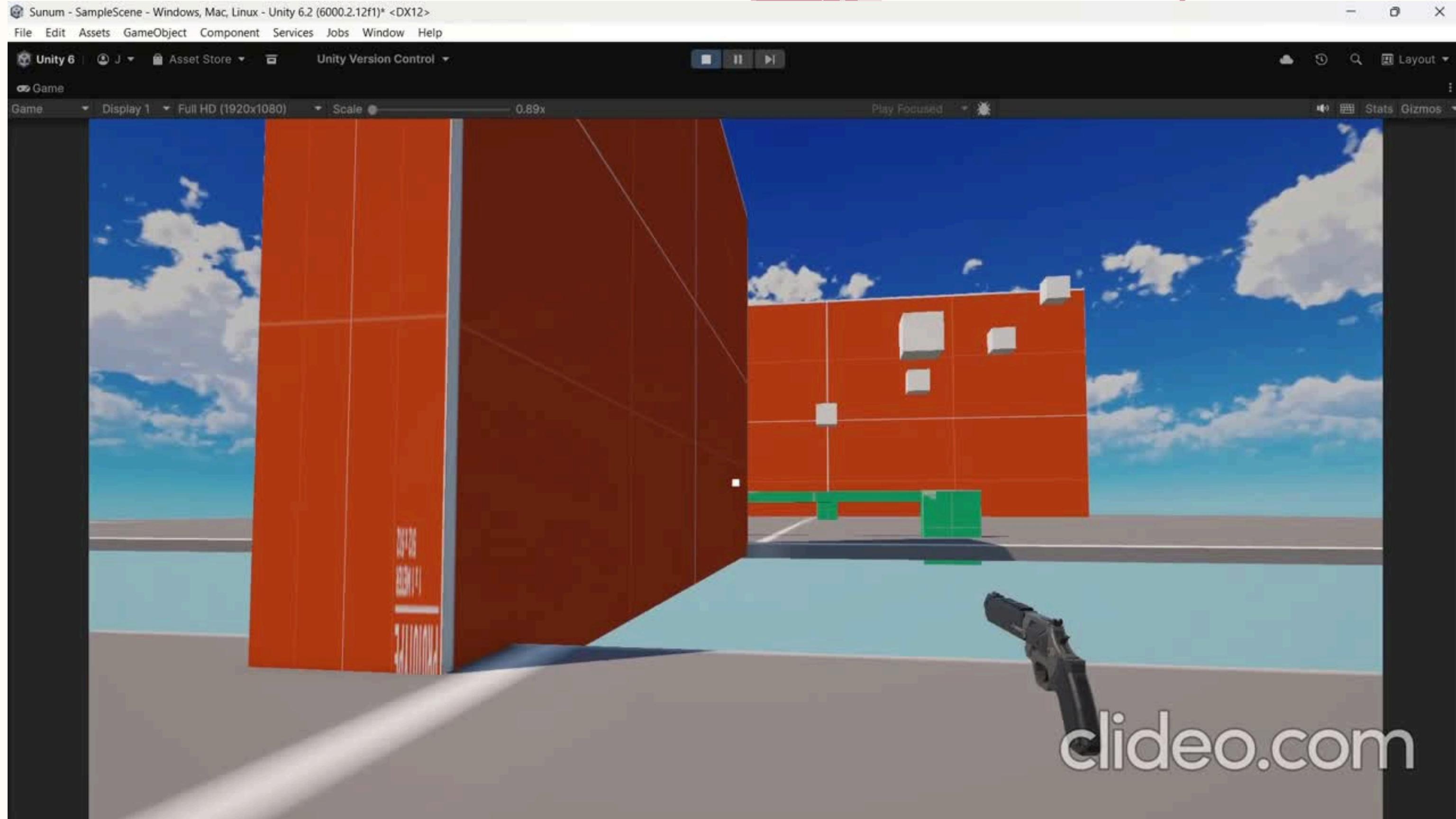
Prompt:

game UI asset, victory popup window frame, sci-fi military interface, dark translucent glass background with glowing orange and neon blue borders, futuristic hud design, empty center for text, cybernetic details, high contrast, stylized vector art, isolated on white background --ar 3:2 --no text

- **Açıklama:**

- **victory popup window frame** : Çerçeve istediğimizi belirttik.
- **sci-fi military interface** : Askeri ama teknolojik (Titan avcısı teması).
- **translucent glass** : Arkası yarı saydam cam (oyun arkada buzlu görünüşün diye).
- **empty center** : Ortayı boş bırak ki yazı yazabilesin.
- **isolated on white background** : Arka planı beyaz yapar, böylece Photoshop'ta veya online araçlarda (remove.bg gibi) o paneli kesip alman çok kolay olur.





clideo.com

Mücadele

Oyunun temel mekanikleri çalışsa da eğlence için bir mücadele olması neredeyse her oyun için şarttır.

Mücadele unsurları:

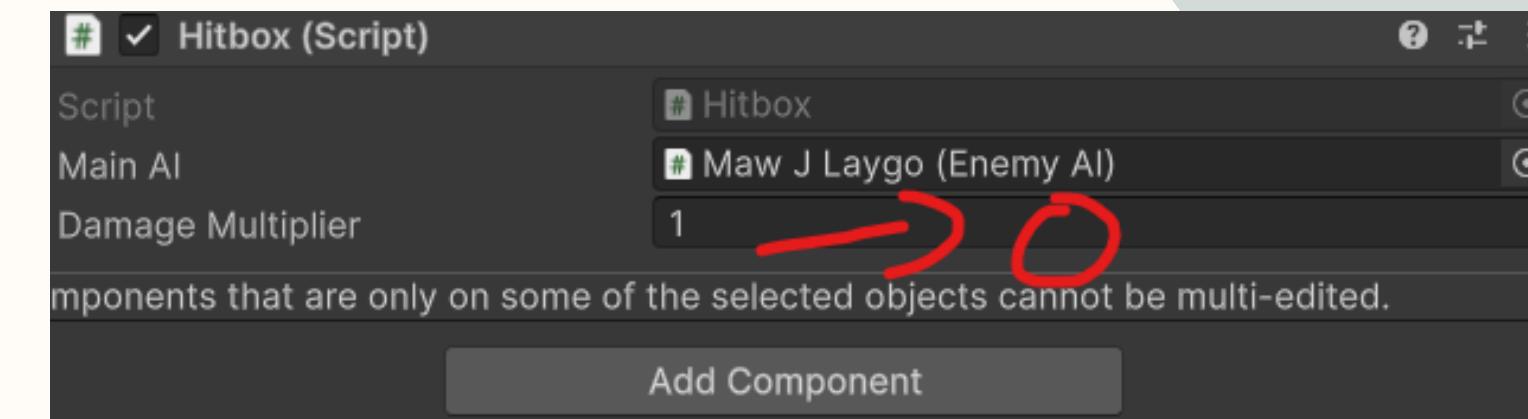
Kaynak Yönetimi

Zorlu Düşmanlar(Dev'in sadece arkasından hasar almasını sağlayarak bölüm tasarımlı değiştirilebilir.)

Engelli parkurlar

Envanter yönetimi vs.

Şu an yapay zeka ile yaptığımız prototip sözde bitmiş durumda ancak bunlardan hiçbir bulunmuyor dolayısıyla bunu tespit etmeliyiz ve eklemeliyiz.



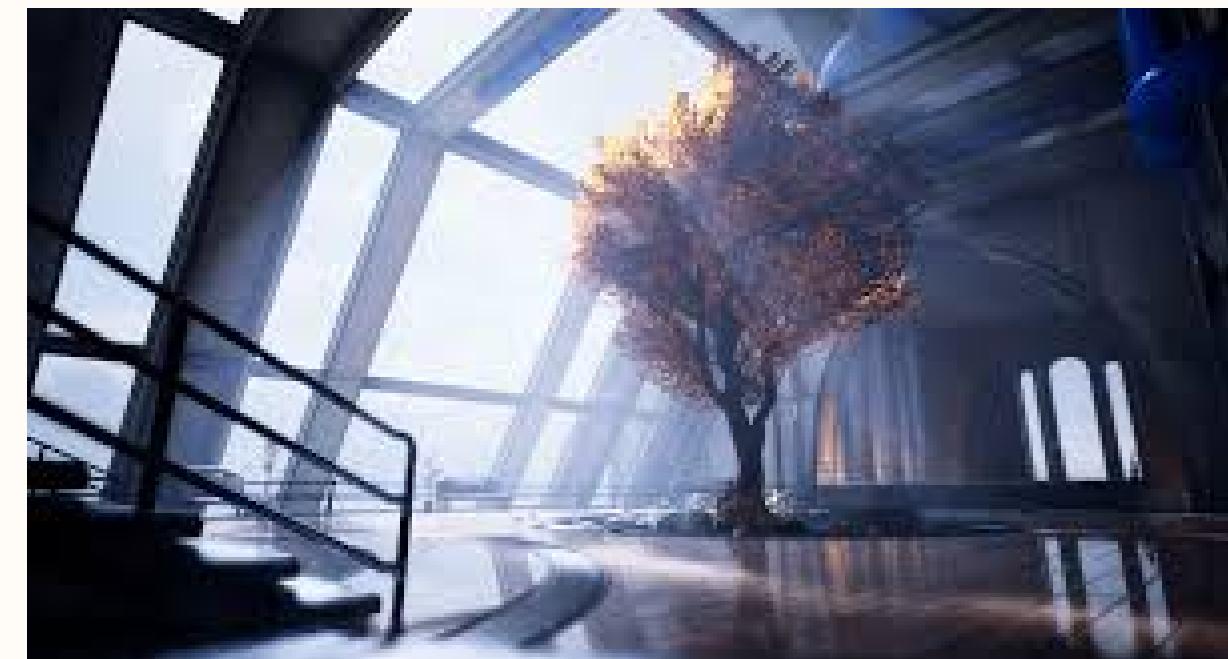
Mücadele

Şimdi Unity'de oyunumuza sınırlı mermi mekanığı ekleyerek, mücadele hissini arttırmayı deneyelim. Bu kaynak yönetimi sayesinde parkurları daha anlamlı hale getirebiliriz.

Soslar

Şu ana kadar ortaya çıkan prototipin eğlenceli olduğuna ikna olduysak, **hikaye, iyi görseller, ses tasarıımı, ışıklandırma** gibi kaliteyi arttıracı unsurlara geçebiliriz.

Ortaya çıkan şeyin eğlenceli olduğunu düşünmüyorsak bu işlere girmeyecek ve projeye devam etmeyeceğiz. Bu kötü görünse de belki de boşça vakit kaybetmeyi önlemiş olabilir.



Teşekkürler!