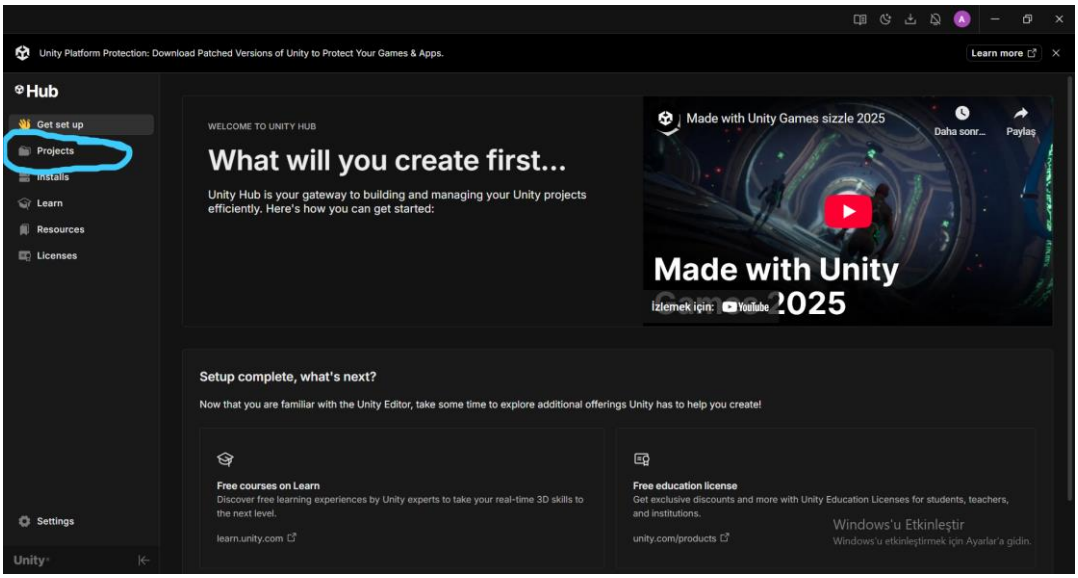
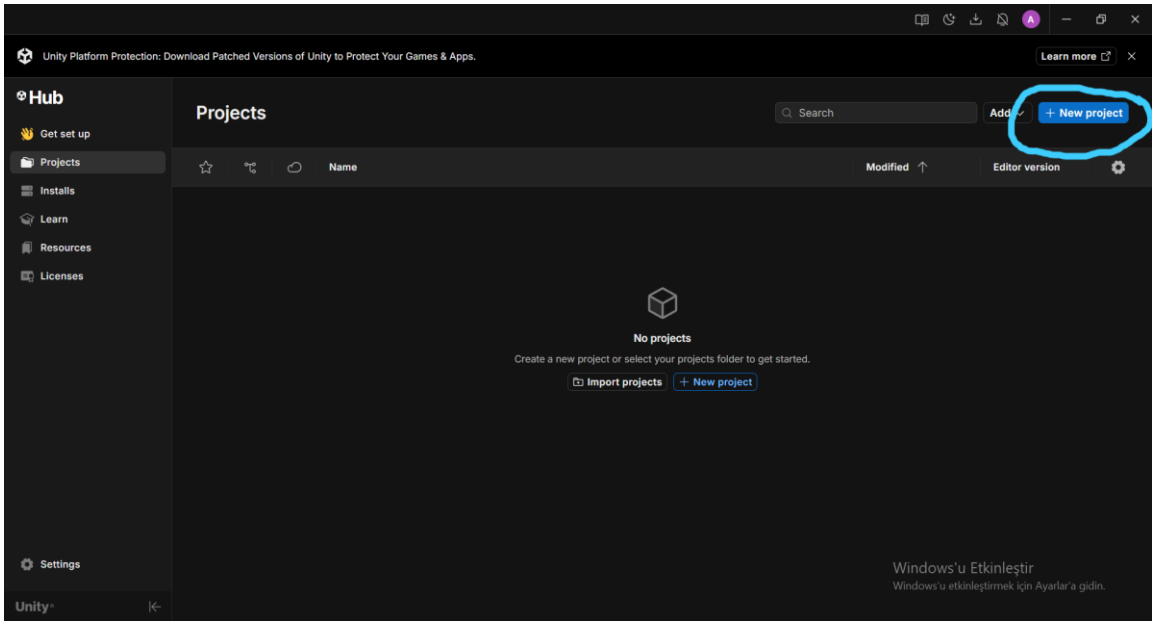
Oyun Programlama Dersi Ödev 1

1) Unity ile Yeni Proje Oluşturma

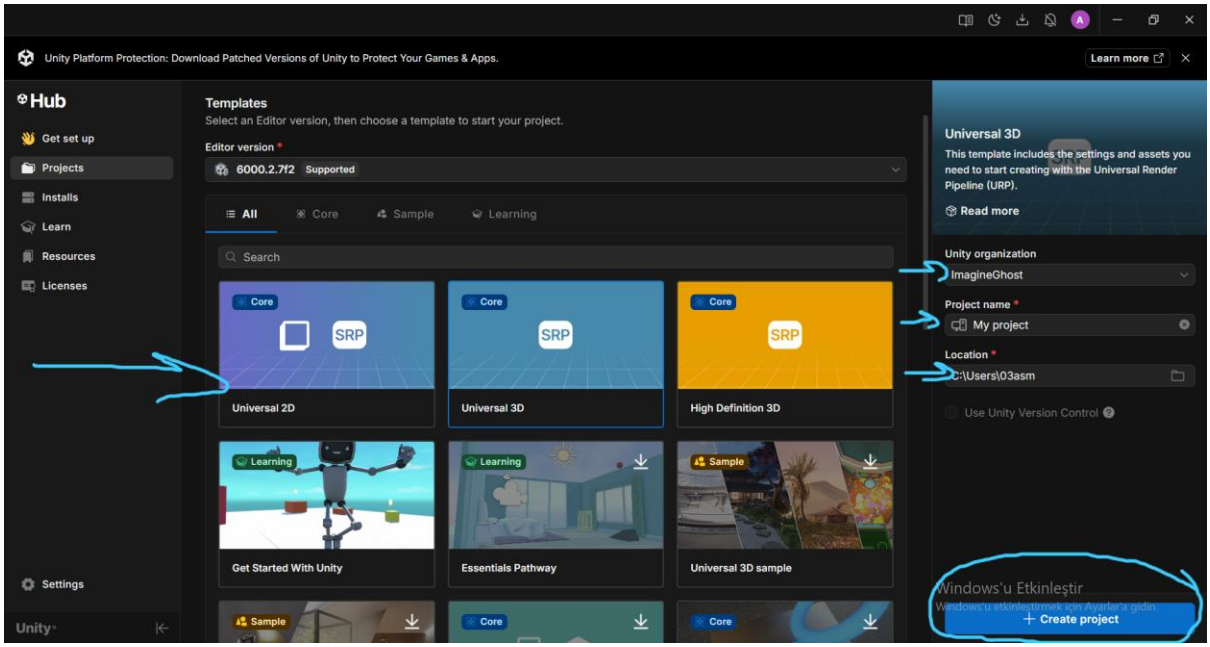
Unity ile yeni ir oyun projesi oluşturmak istediğimizde ilk yapmamız gereken Unity uygulamasını açtıktan sonra ekranın solundaki “Projects” bölümünü açmaktır.



Ardından açılan kısım daha önce proje yapıldıysa onlara ulaştığımız, yapılmadıysa veya yeni yapılacaksa oluşturduğumuz yerdir. Yeni bir proje oluşturmak için açılan kısmın sağ tarafındaki “New Project” butonuna basılmalıdır

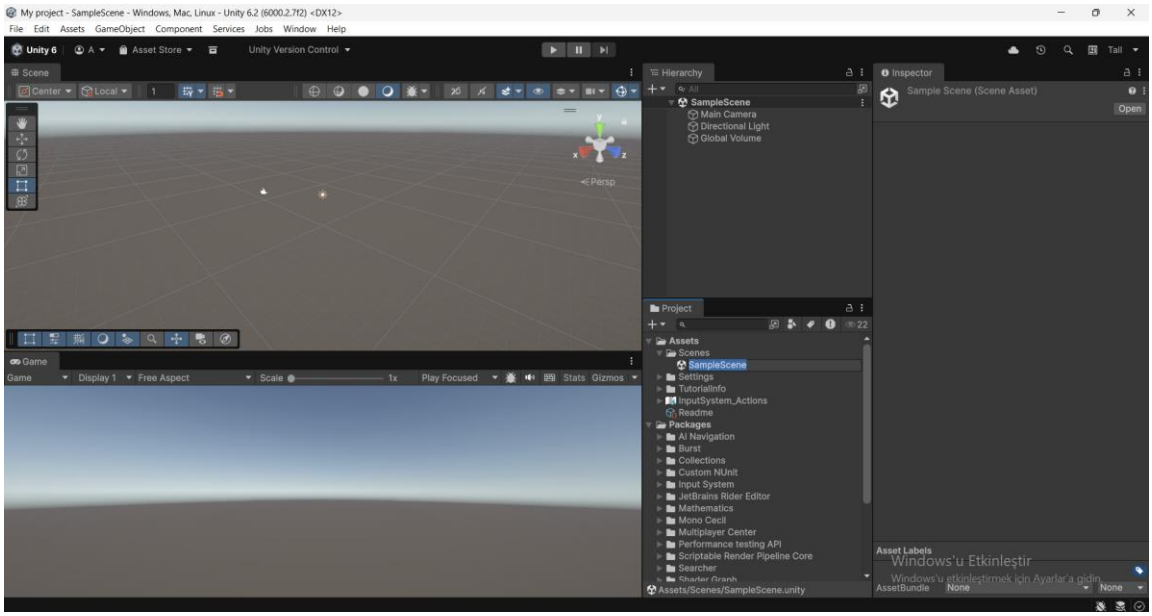


Sonrasında projenin türünü, adını, nereye kaydedileceği gibi özellikleri seçeceğimiz ekran açılacak. İstenilen özellikler seçildikten sonra sağ aşağıdaki “Create Project” butonuna basılmalıdır.

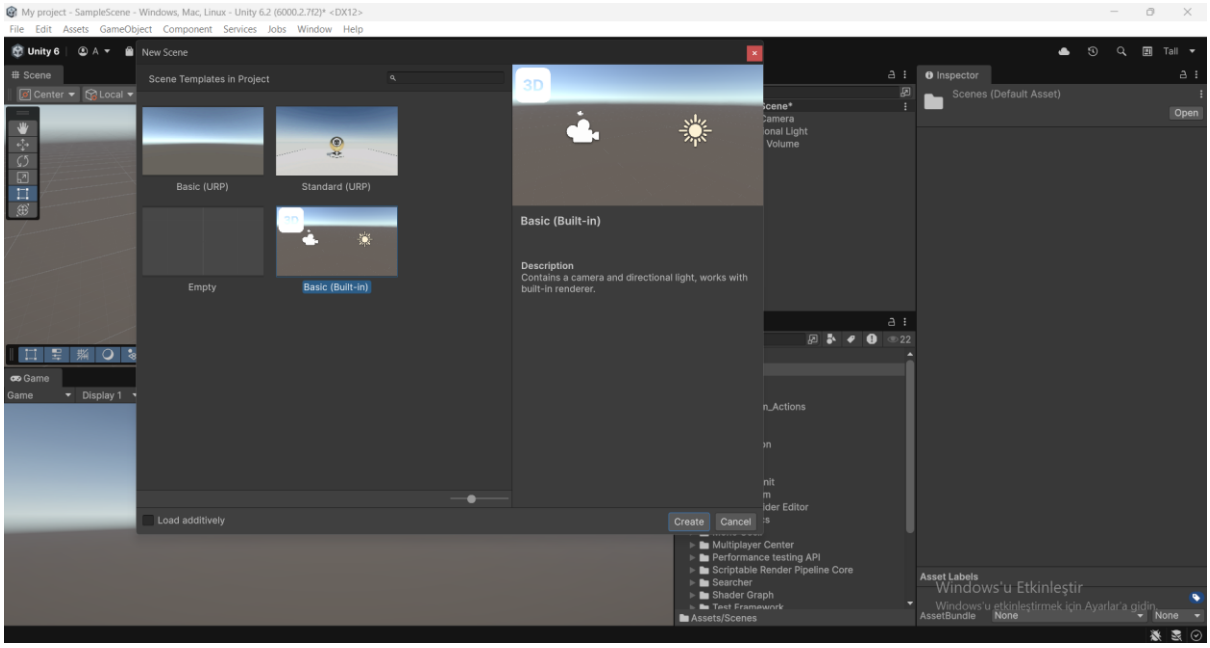


2) Sahne Ekleme ve Silme

İstenilen özellikler seçilip “Create Project” butonuna basıldıktan bir süre sonra yeni proje oluşturulur. Ekranın “Hierarchy” kısmında bulunan “SampleScene”in yanındaki küçük oka basıldığında SampleScene içerisinde olanlar altta gözükür. Açılan kısımda bulunan Main Camera, Directional Light gibi seçeneklere basıldığında seçilen seçeneğin özellikleri sağ taraftaki “Inspector” kısmında gözükür.

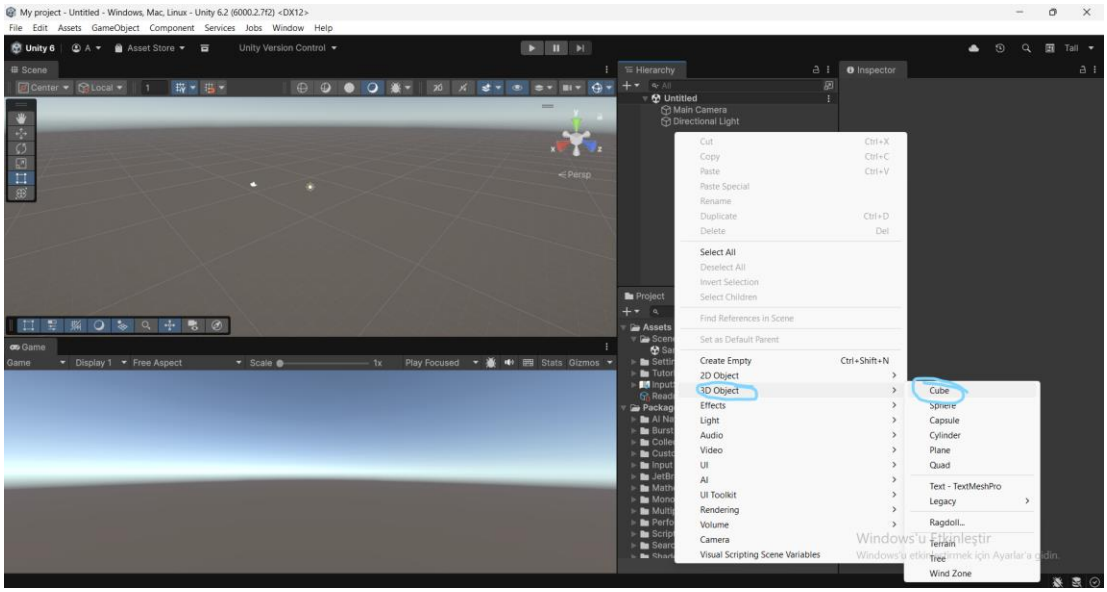


Sahne ayarı için default olarak verilmiş SampleScenes sahnesi Project ekranındaki Scenes kısmından silinir. Sonrasında ekranın sol üst kısmındaki “File” butonundan yeni sahne oluşturulur ve açılan pencerede yeni sahnenin özellikleri seçilir. Bu bizim için “Basic (Built-in)”dir.



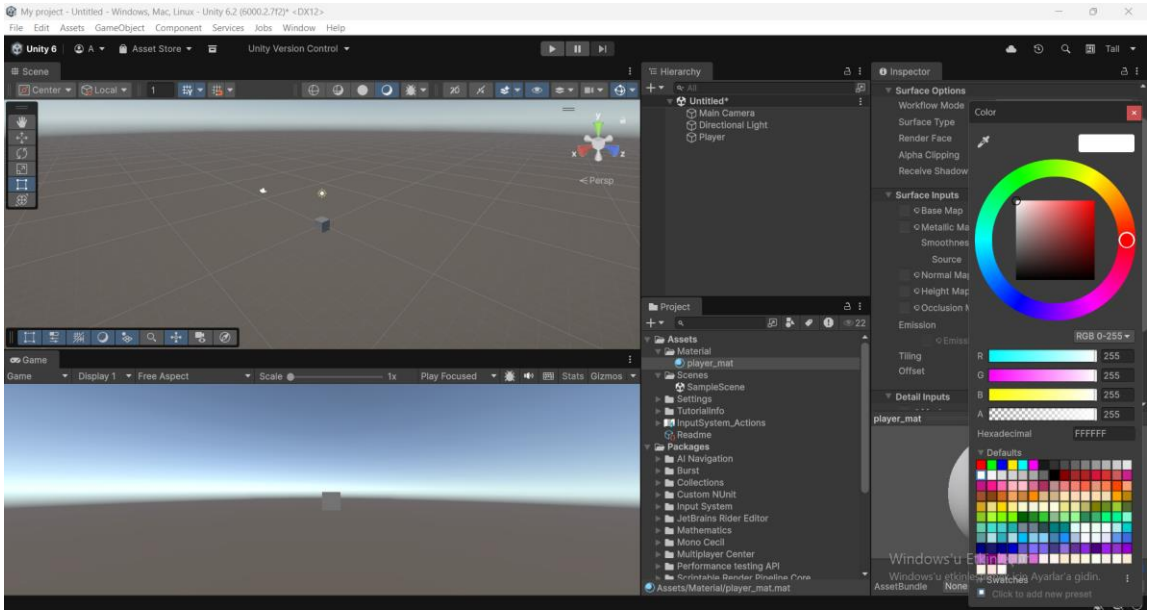
3) Sahneye Nesne Ekleme

Projenize karakter eklemek için Hierarchy kısmında sağ tık yapıp “3D Object” kısmından “Cube” seçtiğimizde sahneye bir küp nesnesi ekleyecektir. Daha sonrasında bu nesneye isim verebilirsiniz. Bizim için bu nesnenin adı “Player”dır.

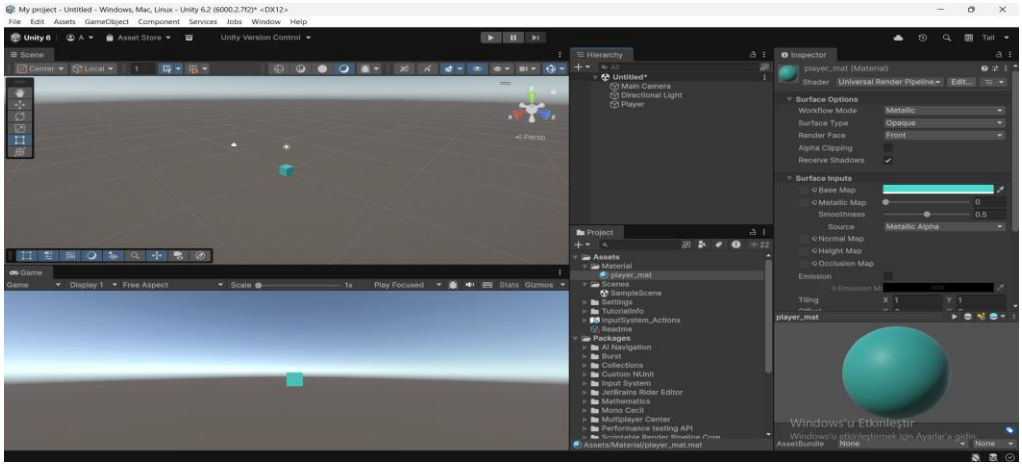


4) Nesneye Materyal Ekleme

Eklenilen nesneye renk gibi materyal ekleyerek eklediğiniz özellikleri bu nesne üzerinde görebilirsiniz. Bunun için Project penceresi içinde sağ tık yapılıp create seçeneği üzerine getirerek folder seçeneği ile yeni bir dosya oluturulmalı. Bu dosyanın ismi Material koyulabilir. Oluşturulan dosyaya sağ tık yapılıp Create kısmından Material seçeneği seçildiğinde materyal oluşturulur. Oluşturulan materyalin üzerine basıldığında Inspector ekranında materyalin özelliklerini görürüz. Renk özelliğini Surface Inputs kısmının altında bulunan Base Map kısmından değiştirebilriz.

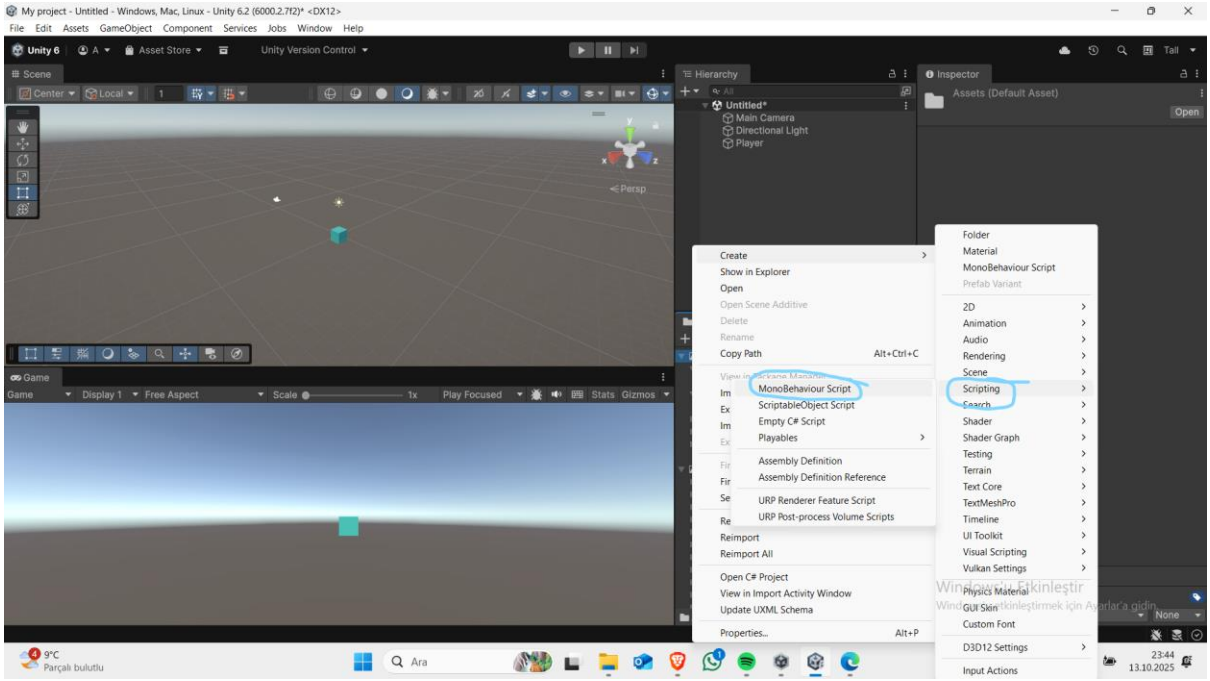


Hazırlanan materyali nesneye eklemek için Project ekranındaki Material dosyası altından Hierarchy ekranındaki oluşturduğunuz nesnenin üzerine sürükleyip bırakılmalıdır.



5) Sahnedeki Nesneye Script ekleme

Oluşturulan nesneye Script eklemek de materyal eklemeye benzer. Project dosyası içerisinde sağ tıklayıp create kısmından folder seçilir ve script içerisinde olacağı dosya oluşturulur. Daha sonra script dosyası içerisinde sağ tıklanır ve create kısmından scriptingin üzerine gelinir ve MonoBehaviour Scripting seçilir. Scripte isim verebilirsiniz

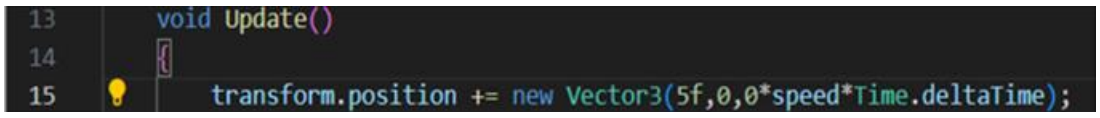


6) Scipt le Nesnenin Konumunu Bir Kez Değiştirme

Oluşturulan karakterin konumunu script dosyasında üzerinde değişiklik yaparak da değiştirebiliriz. Karakterimizin konumunu bir kez değiştirmek istiyorsak script dosyasındaki start() fonksiyonunu aşağıdakine benzer bir kod eklememiz karakterin hareketini sağlayacaktır.

7) Script le Nesnenin Konumunu Sürekli Değşitirme

Oluşturduğumuz karakterin sahne içerisindeki pozisyonunu sadece bir defa değil, sürekli olarak değiştirmek istiyorsak, bu işlemi update() fonksiyonu içinde yapmamız gerekir. Böylece karakterimiz her karede frame pozisyonunu günceller ve sürekli hareket ediyormuş gibi görünür. Bu durum, konumu bir kez değiştirmeye benzer şekilde gerçekleştirilir ancak bu kez update() fonksiyonunun sürekli çalışması sayesinde hareket devamlı hale gelir.



8. Zamanın Normalzasyonu

Update() fonksiyonu içinde yazılan hareket kodu, her bilgisayarda çalıştırıldığında donanım gücüne bağlı olarak farklı performanslar gösterebilir. Daha güçlü bir bilgisayarda kare hızı fps yüksek olacağı için nesne daha hızlı hareket eder, bu da oyuncular arasında farklı oyun deneyimlerine neden olur. Bu farkın sebebi, update() fonksiyonunun saniyede çalıştırılma sayısının cihazın donanımına göre değişmesidir. Bu nedenle fps farklarından kaynaklanan hız farklılıklarını ortadan kaldırmak için hareket

kodu Time.deltaTime ile çarpılır. Time.deltaTime, iki kare arasındaki süreyi temsil eder ve bu sayede hareket hızı donanımdan bağımsız hale gelir. Böylece nesne, her bilgisayarda aynı hızda hareket eder. Kullanıcı şartları eşitlenir.

9. Speed Değişken Tanımlama ve Tanımlama Farkları

Karakterimizin hareket hızı varsayılan olarak sabit bir değere sahiptir. Ancak speed isimli bir değişken tanımlayarak karakterin bir saniyede ne kadar mesafe kat edeceğini belirleyebiliriz. Bu değişkeni sınıf içinde tanımlayıp istediğimiz bir değer atarız ve hareket kodunu bu değişkenle çarparak nesnenin hareket hızını kontrol edebiliriz. Geliştirme sürecinde bu değişkenin public olarak tanımlanması önerilir, çünkü bu sayede Inspector panelinde görünebilir ve test aşamasında kolayca değiştirilebilir. Fakat public olarak tanımlanan değişkenler dışarıdan da değiştirilebilir. Bu nedenle oyun tamamlanmaya yaklaştığında değişkenlerin private yapılması önerilir. Private olarak tanımlandığında değişken Inspector ekranında görünmez ve dışarıdan erişilemez.

```
// Start is called once before the first execution of Update after the MonoBehaviour is created
1 reference
public float speed = 3;
```

```
5    // Start is called once before the first execution of Update after the MonoBehaviour is created
     1 reference
6    private float speed = 3;
```

10.Klavyeden Yön Tuşları ile Nesnenin Hareketinin Kontrolü

Karakterin hareketini sadece kodla değil, klavyeden yön tuşlarına basarak da kontrol etmek mümkündür. Bunun için update() fonksiyonu içindeki hareket kodunda bazı değişiklikler yapılır. Hareketin x ve y eksenlerinde klavyeden alınan girdilere göre belirlenebilmesi için Vector3 içinde bulunan x ve y değerleri Input.GetAxis() fonksiyonu ile değiştirilir. Input.GetAxis fonksiyonu, kullanıcının bastığı tuşlara göre -1 ile +1 arasında bir değer döndürür. X ekseni için Input.GetAxis("Horizontal"), Y ekseni için ise Input.GetAxis("Vertical") kullanılır. Bu ifadeler Unity tarafından varsayılan olarak

tanımlanmıştır ancak istenirse ayarlar kısmından değiştirilebilir. Bu şekilde karakterin klavyeden gelen girdilere göre yatay ve dikey eksenlerde hareket etmesi sağlanır.

Projenin script dosyasına github hesabımdan ulaşabilirsiniz:

https://github.com/AsmBrk/Oyun-Programlama-Lab