Kodla animasyon nasıl oluşturulur?

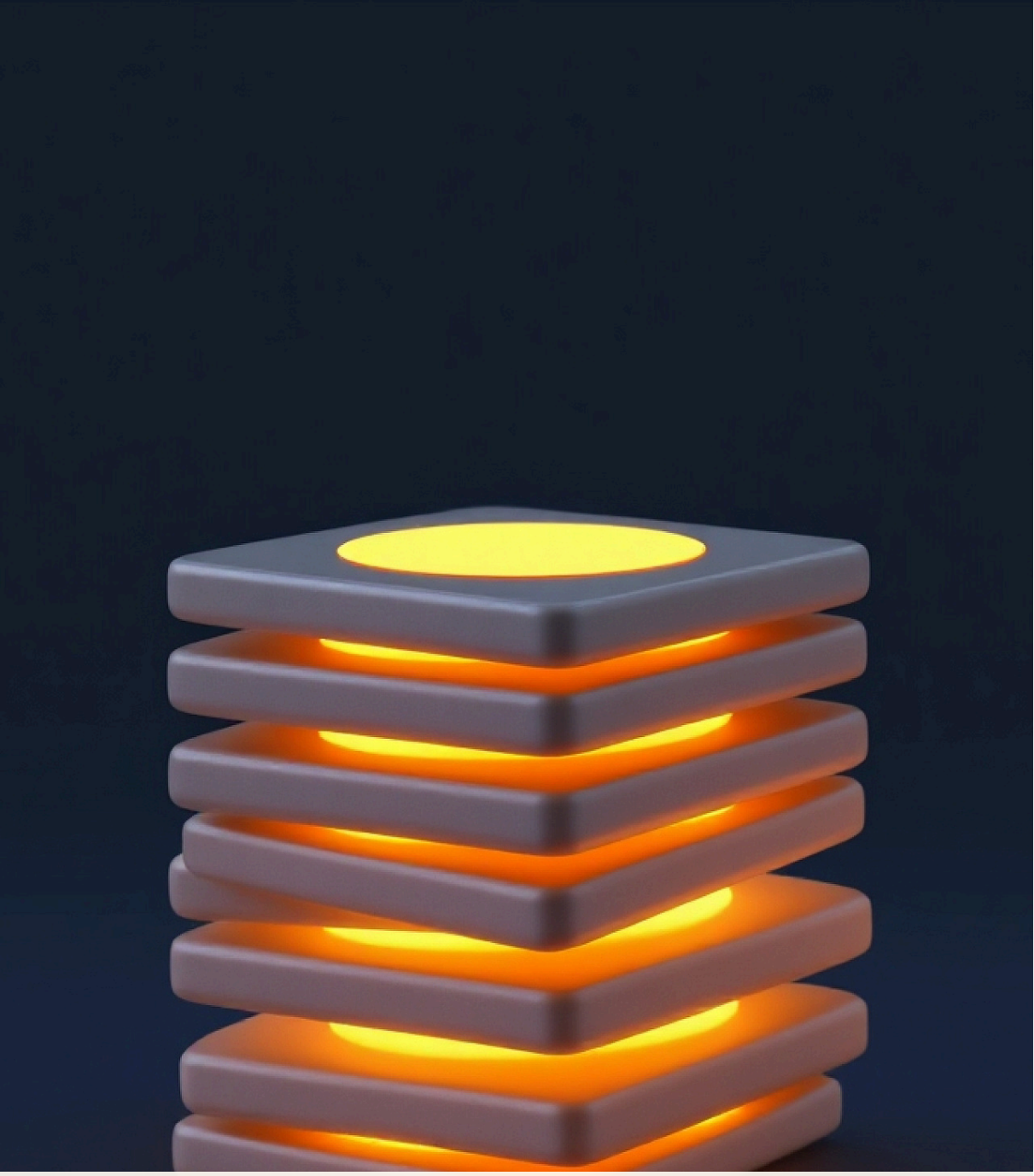


www.yzokul.com.tr

Kodla animasyon nasıl oluşturulur?



Kod bir animasyonu nasıl yapar? Kısa cevap şudur: kod, bir görüntünün özelliklerini zaman içinde değiştirir ve gözün bu değişimi hareket olarak algılar. Özellikler, görüntünün nerede olduğu, ne kadar büyük olduğu veya hangi renkte olduğu gibi şeylerdir. Bunlar adım adım değiştiğinde, resim hareket ediyormuş gibi görünür. İşin içinde sihir yoktur; sadece tekrarlanan değişiklikler vardır.



Bir animasyon, özellikleri sürekli değişen bir resimden ibarettir. Aynı görüntü her seferinde küçük farklılıklarla tekrar tekrar çizilir. Gözün bu hafifçe farklı resimleri birleştirerek pürüzsüz hareket olarak algılar. Sihirli bir şey olmaz; bilgisayar sadece resmi yeni değerlerle yeniden çizer. Bu tekrarlanan yeniden çizim, tüm kod animasyonunun özüdür.



Kod bir görüntünün birkaç özelliğini değiştirebilir: ekrandaki konumu, boyutu, rengi ve hangi resmin gösterildiği. Bunlardan yalnızca birini değiştirmek bile görünür bir animasyon yaratabilir. Örneğin, bir nesnenin konumunu değiştirmek onun kaymasını sağlarken, boyutunu değiştirmek onu büyütür. Daha zengin efektler için bunları birleştirebilirsin.



Bir nesnenin sağa kaymasını sağlamak için kod, resim her çizildiğinde yatay konumunu küçük bir miktar artırır. Örneğin, 0 konumundan başla ve her karede 5 ekle. Nesne ekranda pürüzsüzce hareket ediyormuş gibi görünür. Yatay konuma genellikle x koordinatı denir. Bu basit toplama, kayma hareketini yaratır.



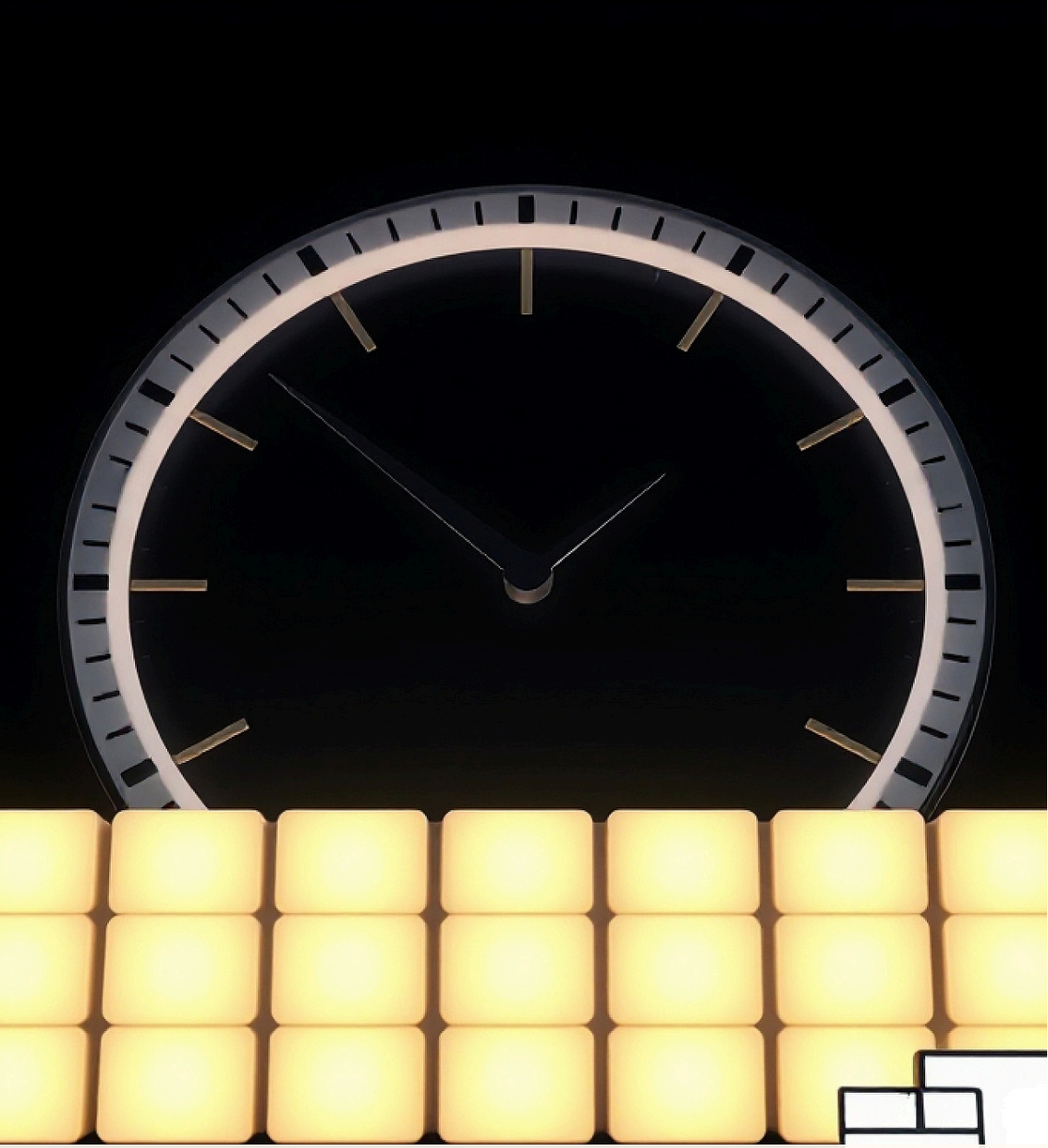
Bir nesnenin boyutunu adım adım değiştirmek onu büyütür veya küçültür. Örneğin, her karede genişliğini ve yüksekliğini 2 piksel artır. Rengini kademeli olarak değiştirmek, örneğin her seferinde biraz kırmızı eklemek, tonunu kaydırır. Bu özellik değişiklikleri, ekranda herhangi bir hareket olmasa bile animasyon olarak algılanır. Bunlar, hareketin tek animasyon türü olmadığını gösterir.



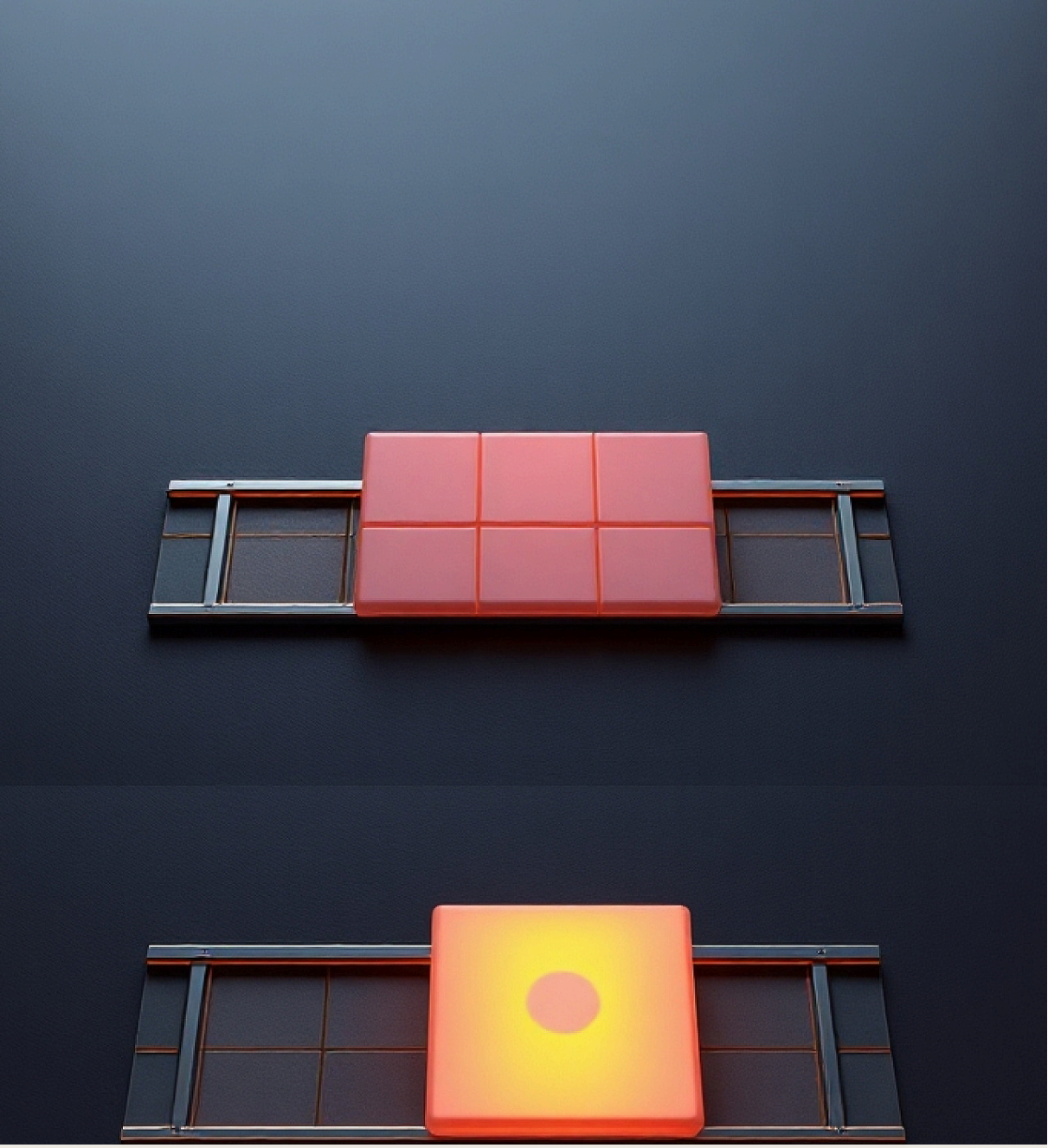
Animasyon, birbiri ardına gösterilen bir dizi hareketsiz resimdir. Her hareketsiz resme kare denir. Kareler hızlı gösterildiğinde, gözün bunları sürekli hareket olarak algılar. Bir flipbook düşün: her sayfa bir karedir ve hızlı çevirmek hareket yaratır. Kod, resmi güncellenmiş özelliklerle çizerek kareler oluşturur.



Bir karakterin hareket ediyormuş gibi görünmesini sağlamak için kod, o karakterin farklı çizimlerini sırayla gösterir. Örneğin, bir bacağı kalkmış, sonra düz, sonra tekrar kalkmış göster. Her çizim bir karedir. Sırayla hızlı gösterildiğinde, karakter yürüyormuş gibi görünür. Kod, her karede hangi resmin gösterileceğini basitçe değiştirir.



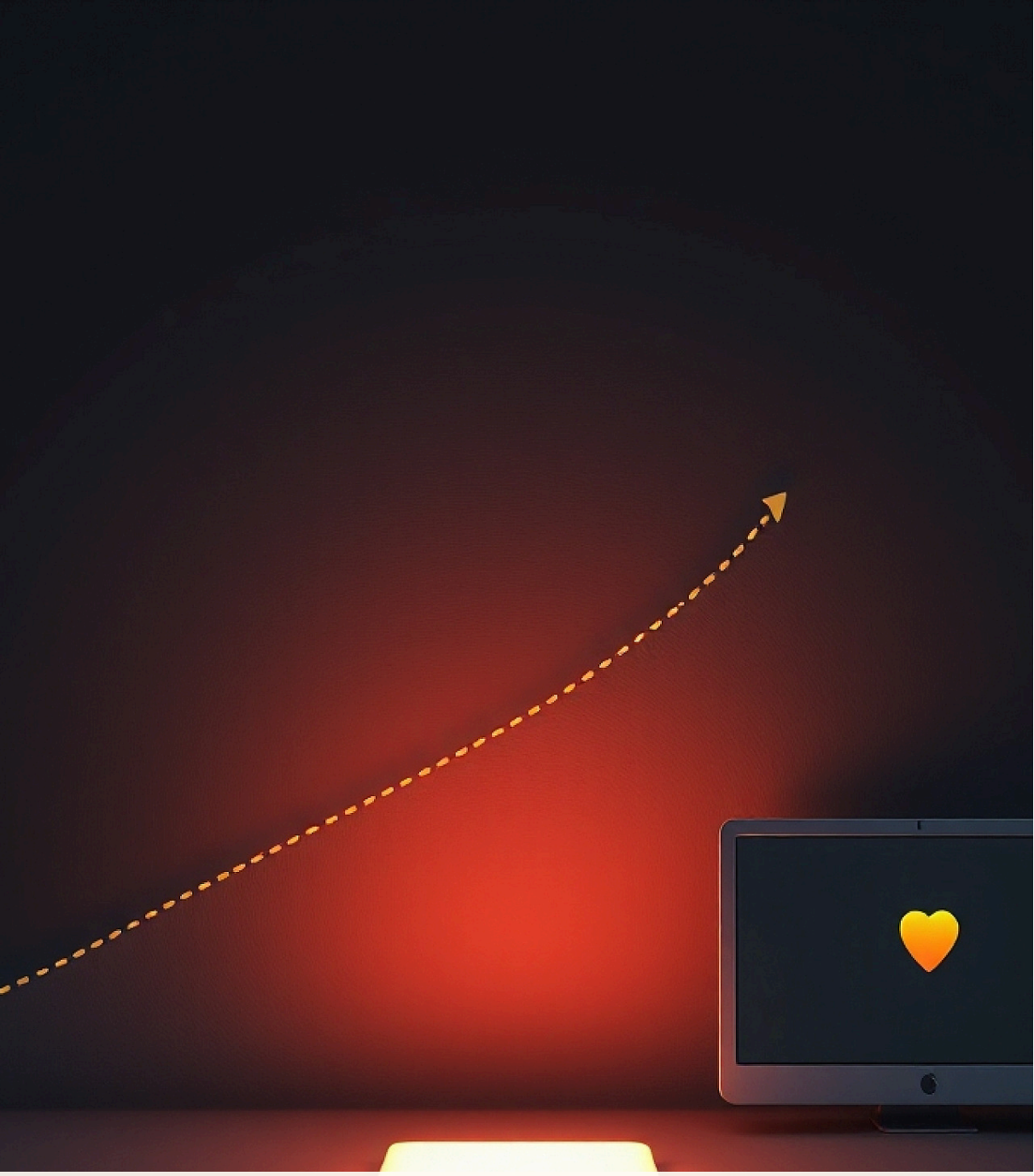
Kareler arasındaki süre çok önemlidir. Kareler çok yavaş gelirse, hareket bir slayt gösterisi gibi sıçramalı görünür. Yeterince hızlı, genellikle saniyede birçok kare gelirse, hareket pürüzsüz görünür. Kod, bir sonraki karenin ne zaman çizileceğini kontrol etmek için bir zamanlayıcı veya döngü kullanır. Tutarlı zamanlama, pürüzsüz animasyonun anahtarıdır.



Kare başına daha büyük bir değişiklik daha hızlı hareket, daha küçük bir değişiklik daha yavaş hareket yaratır. Örneğin, her karede konuma 10 eklemek, 5 eklemekten iki kat daha hızlı hareket ettirir. Hız, her seferinde ne kadar değiştirdiğinle belirlenir. Hızı etkilemek için kareler arasındaki süreyi de değiştirebilirsin. Hem adım büyüklüğü hem de zamanlama, animasyonun ne kadar hızlı görüneceğini kontrol eder.



Kod, bir sonraki kareye karar vermek için basit hesaplamalar kullanır. Örneğin, yeni konumu elde etmek için mevcut konuma bir sayı ekler. Bu hesaplamalar hareketin nasıl görüneceğini kontrol eder. Eğriler veya ivme yaratmak için çarpma veya başka matematik kullanabilirsin. Hesaplamalar, animasyonu ileriye doğru sürükleyen kurallardır.



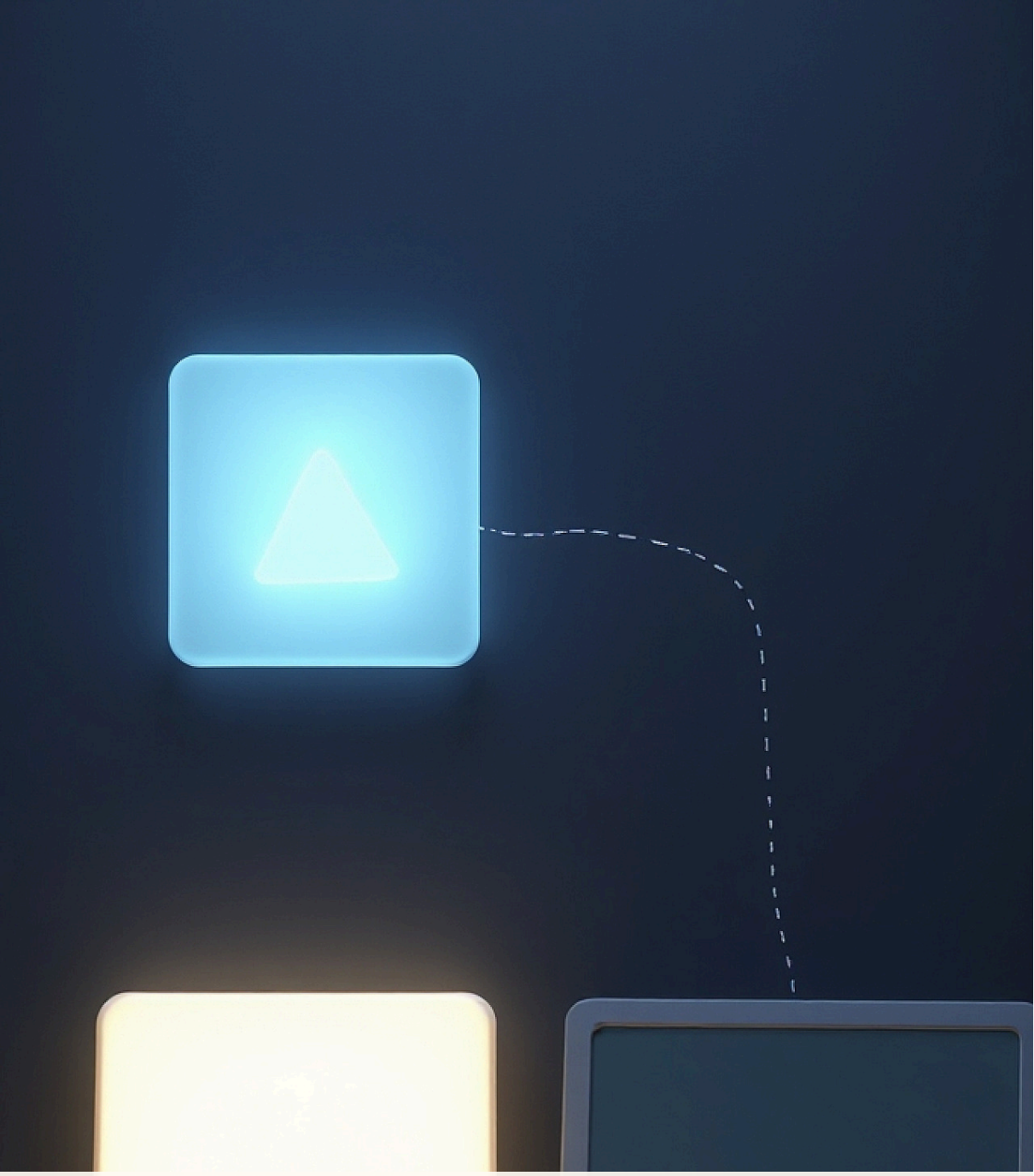
Değerlerin zaman içinde değişme biçimi geçiş stilini verir. Her karede aynı miktarı eklemek gibi sabit bir değişiklik eşit ve doğrusal görünür. Küçük değişikliklerle başlayıp bunları artırmak gibi diğer desenler, hareketin yavaş başlayıp hızlanmasını sağlar. Kod, bu stilleri yaratmak için farklı formüller kullanabilir. Geçiş stili, animasyonun hissini etkiler.



Kod aynı adımları tekrar tekrar yineler: bir özelliği değiştir, resmi çiz, kısa bir süre bekle ve tekrarla. Animasyonu devam ettiren şey bu tekrarlanan döngüdür. Döngü olmadan animasyon tek bir değişiklikten sonra durur. Döngü sürekli çalışır, resmi saniyede birçok kez günceller ve yeniden çizer.



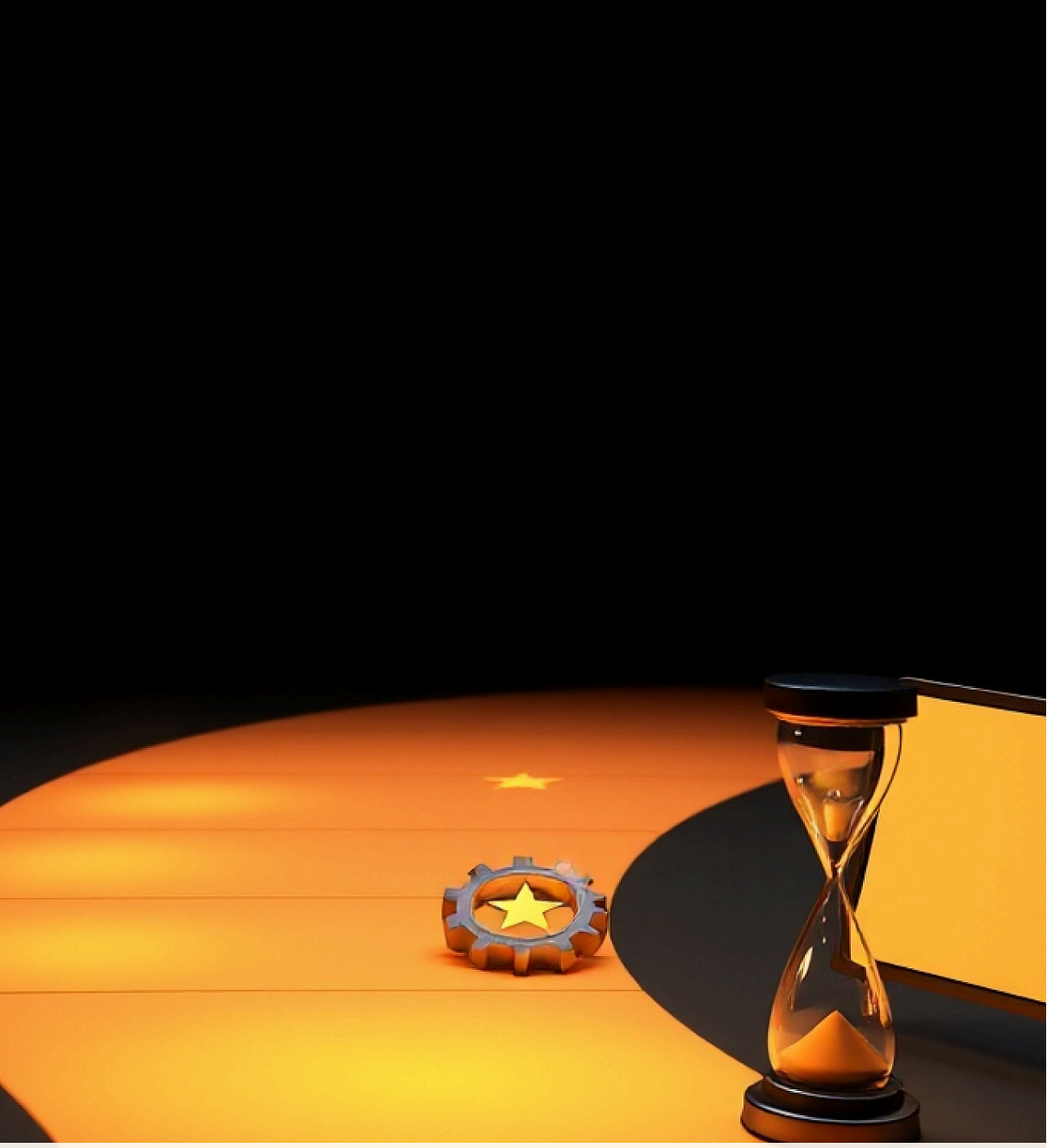
Tam bir örneği takip et: bir nesneyi 0 konumunda başlat. Her karede konumuna 5 ekle, saniyenin 1/60'ı kadar bekle ve yeniden çiz. Nesne sağa doğru pürüzsüzce kayar. Tüm parçaları bir arada görmek—başlangıç değeri, kare başına değişiklik, bekleme ve yeniden çizme—hareketin nasıl ortaya çıktığını gösterir. Kod animasyonunun özü budur.



Yeni başlayanların sık yaptığı bir hata, bir değeri değiştirip resmi bir daha hiç çizmemektir. Örneğin, konum değişkenini güncellersin ama çizim fonksiyonunu çağırmayı unutursun. Yeniden çizim olmadan ekranda hiçbir şey değişmez. Özellikleri değiştirdikten sonra her zaman yeniden çizmeyi unutma. Ekran yalnızca en son çizdiğini gösterir.



Bir başka hata da yanlış zamanlamadır: kareler çok hızlı, çok yavaş veya düzensiz zamanlarda gösterilir. Kareler çok hızlıysa, hareket görülemeyecek kadar hızlı olabilir; çok yavaşsa, sarsıntılı görünür. Düzensiz zamanlama animasyonun bozuk görünmesine yol açar. Kare zamanlamasını kontrol etmek için tutarlı bir zamanlayıcı veya döngü kullan. Pürüzsüz sonuçlar için zamanlamayı test et ve ayarla.



Özetle animasyon, özellikleri zaman içinde değişen ve bir döngü içinde kare kare gösterilen bir resimdir. Zamanlama ve hesaplamalar hareketin hızını ve stilini kontrol eder. Farklı efektler yaratmak için konumu, boyutu, rengi veya gösterilen resmi değiştir. Yeniden çizmeyi ve zamanlamayı tutarlı tutmayı unutma. Bu temellerle kendi kod animasyonlarını oluşturmaya başlayabilirsin.