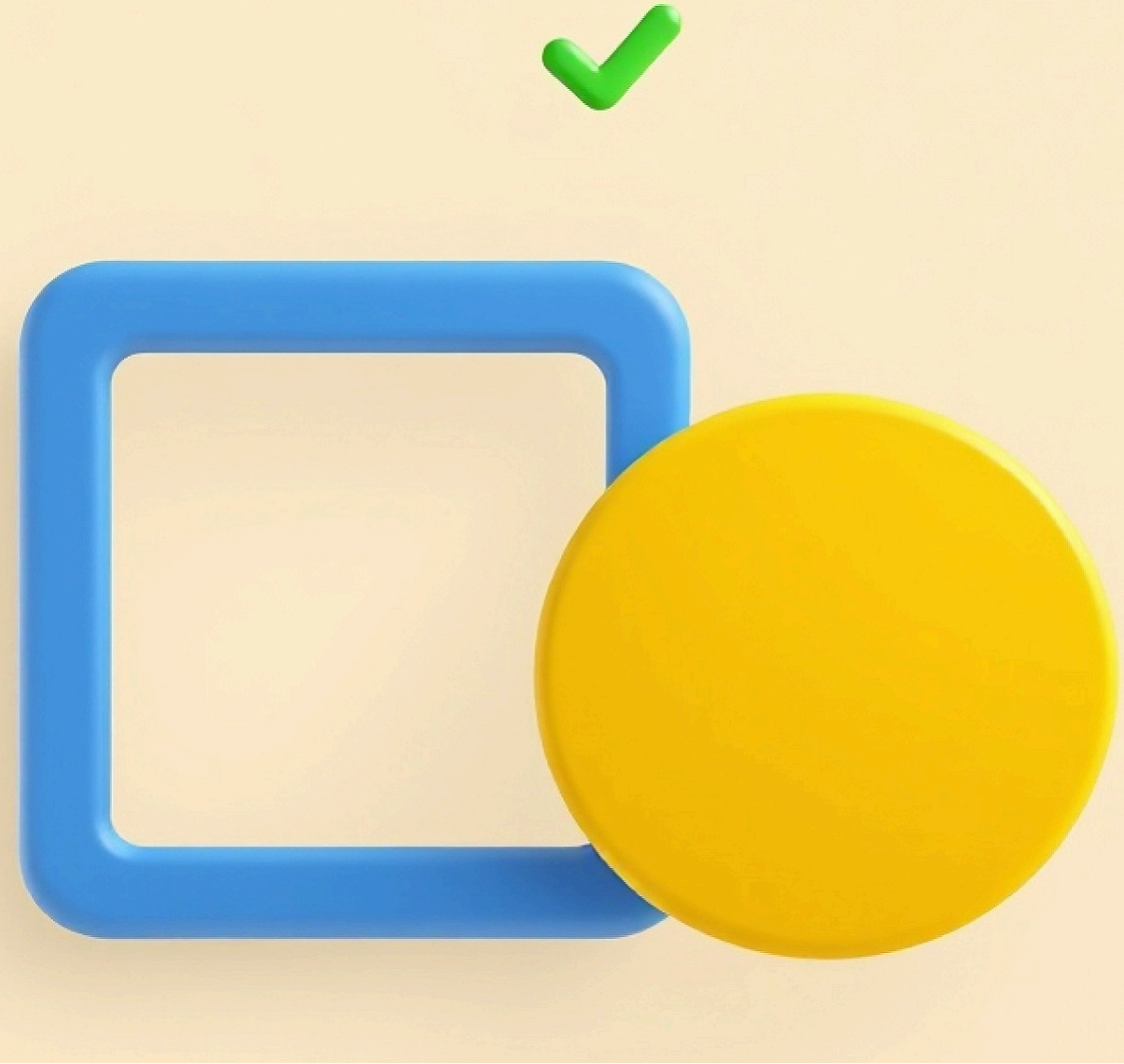
Oyunlarda çarpışma nasıl algılanır?



www.yzokul.com.tr

Oyunlarda çarpışma nasıl algılanır?

www.yzokul.com.tr



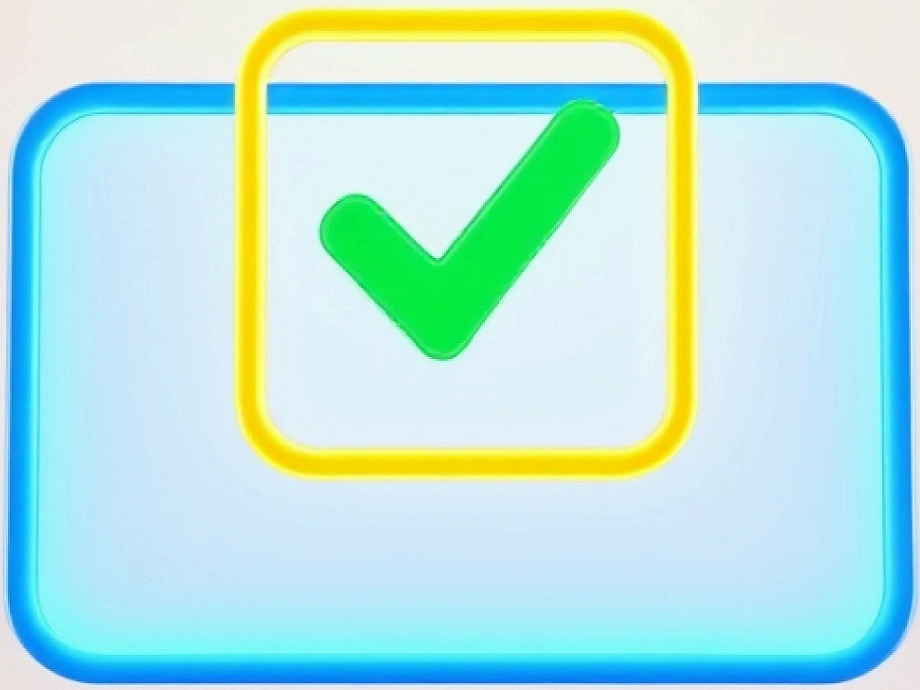
Bir oyunun iki şeyin, örneğin bir karakter ile bir duvarın birbirine dokunduğunu bilmesi gerekir. Kısa yanıt, oyunun nesnelerin nerede olduğunu ve kenarlarının nerede olduğunu karşılaştırmasıdır. Kenarların üst üste gelip gelmediğini kontrol eder. Üst üste geliyorsa oyun bir çarpışma olduğuna karar verir. Bu karşılaştırma saniyede birçok kez yapılır.



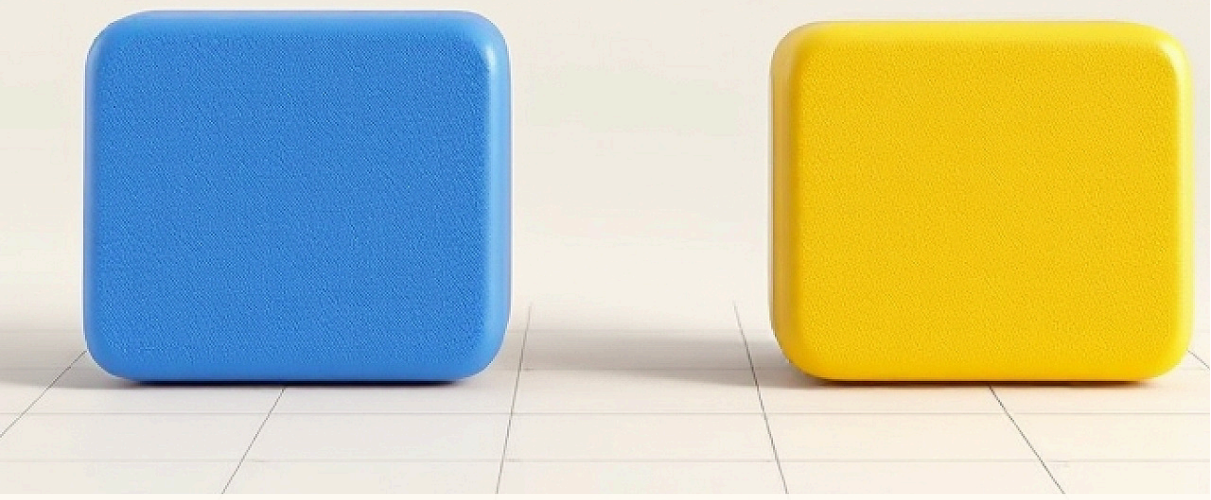
Bir oyundaki her nesnenin, örneğin bir karakterin, bir duvarın veya bir topun, bir çift sayı olarak saklanan bir konumu vardır. Bu sayılar oyuna nesnenin ekranda nerede olduğunu söyler. Örneğin, bir karakter (100, 50) konumunda olabilir; bu, soldan 100 piksel ve üstten 50 piksel anlamına gelir. Nesneler hareket ettikçe oyun bu sayıları günceller.



Her nesnenin ayrıca bir sınırı vardır; bu, dış kenarını işaretleyen bir şekildir. Sınır genellikle nesnenin gerçek görüntüsünden daha basittir. Oyun, iki nesnenin üst üste gelip gelmediğine karar vermek için bu sınırları kullanır. Örneğin, bir karakterin çevresinde dikdörtgen bir sınır olabilir. Sınır, oyunun çarpışmaları hızlıca kontrol etmesine yardımcı olur.



Çarpışma algılama, bir nesnenin sınırını başka bir nesnenin sınırıyla karşılaştırmak anlamına gelir. İki sınır üst üste gelirse oyun bir çarpışma olduğunu söyler. Bu karşılaştırma, her sınırın konumu ve boyutu kullanılarak yapılır. Örneğin, oyun karakterin sınırının duvarın sınırıyla üst üste gelip gelmediğini kontrol eder. Evet ise bir çarpışma algılanır.



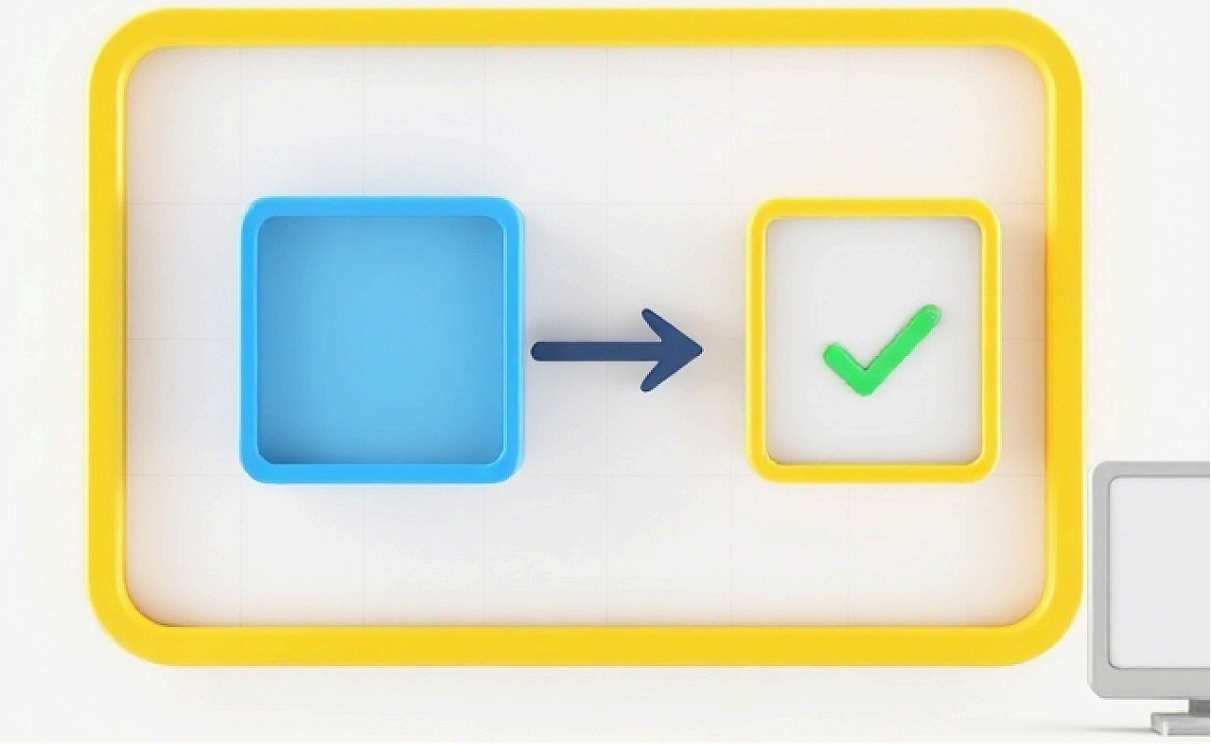
En basit sınır şekli dikdörtgendir. Birçok basit oyun her nesneyi bir dikdörtgenle sarar ve iki dikdörtgenin üst üste gelip gelmediğini kontrol eder. Bir dikdörtgen, konumu ile genişliği ve yüksekliğiyle tanımlanır. Örneğin, bir karakter genişliği 30 ve yüksekliği 50 olan bir dikdörtgen olabilir. Oyun daha sonra bu dikdörtgenin bir duvarın dikdörtgeniyle üst üste gelip gelmediğini kontrol eder.



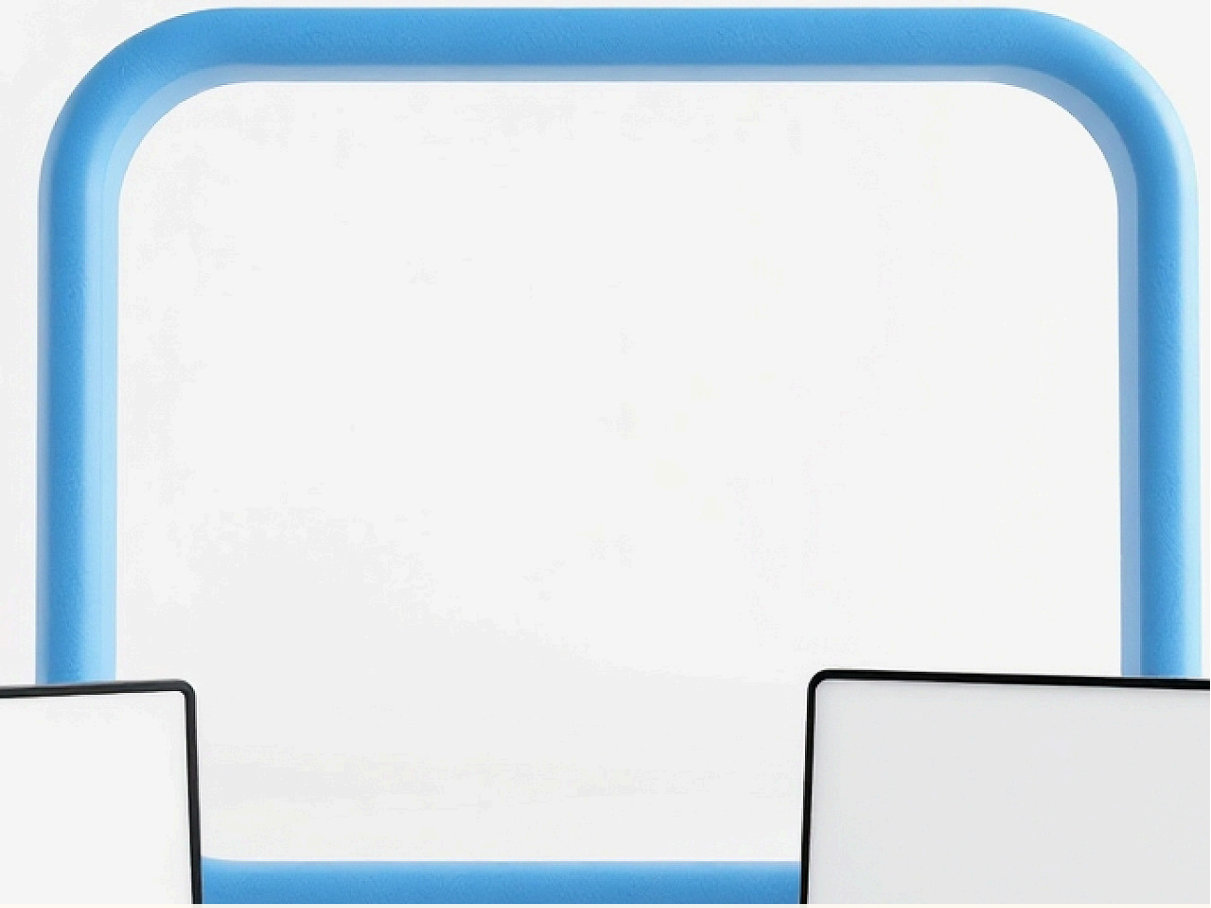
İki dikdörtgenin üst üste gelip gelmediğini kontrol etmek için oyun ekrandaki kenarlarını karşılaştırır. Bir dikdörtgenin sol kenarının diğerinin sağ kenarının solunda olup olmadığını ve benzer şekilde üst ve alt kenarlar için de kontrol eder. Bu koşulların tümü doğruysa dikdörtgenler bir alanı paylaşır ve oyun bunu bir çarpışma olarak ele alır. Örneğin, bir karakterin dikdörtgeni ile bir duvarın dikdörtgeni herhangi bir alanı paylaşıyorsa bir çarpışma algılanır.



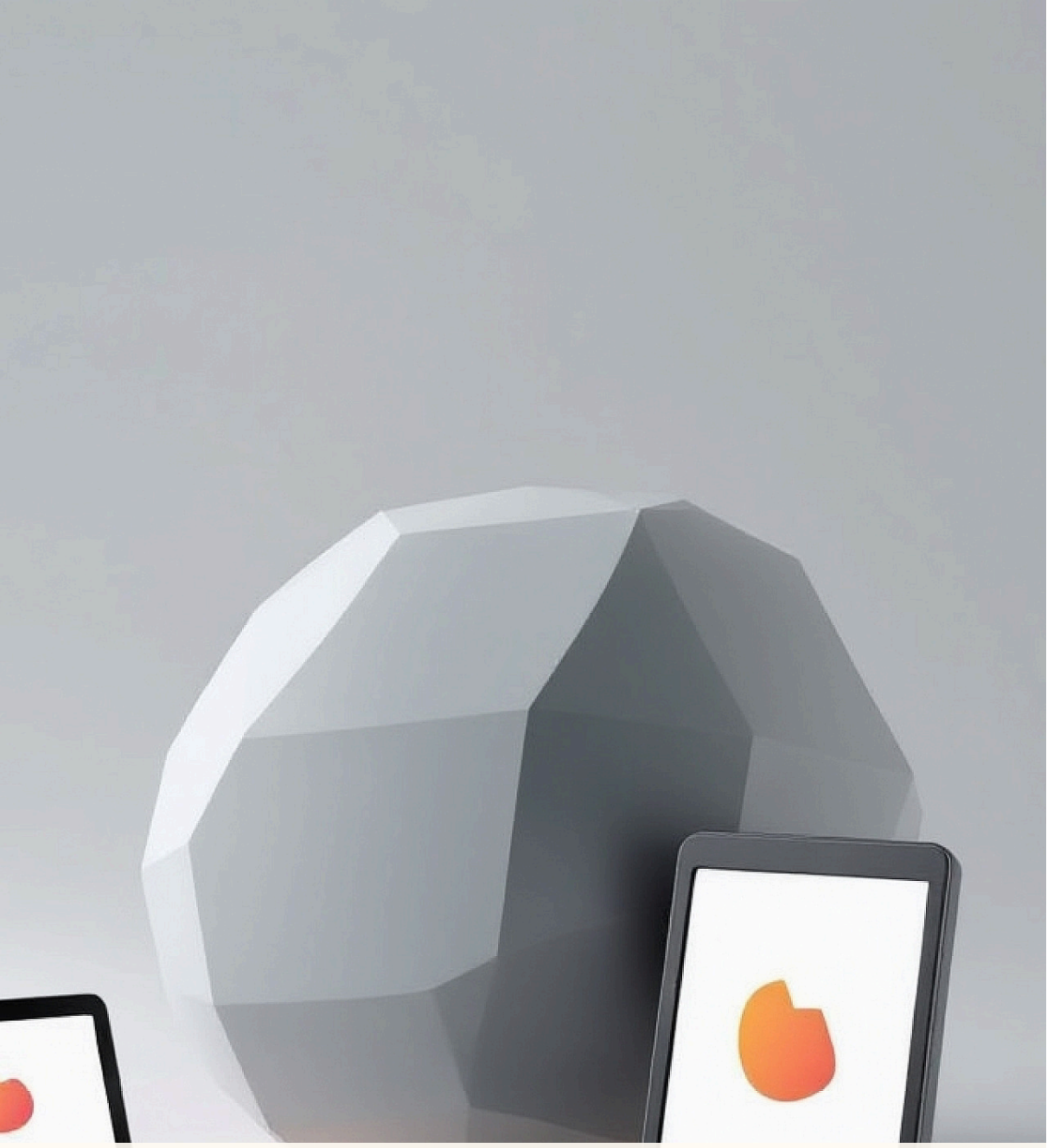
Daireler bir başka kolay sınır şeklidir. Oyun, iki dairenin kenarlarının dokunacak veya kesişecek kadar yakın olup olmadığını kontrol eder. Bu, merkezleri arasındaki mesafenin yarıçaplarının toplamıyla karşılaştırılmasıyla yapılır. Mesafe toplamdan küçükse daireler üst üste gelir. Örneğin, bir oyundaki iki topun dairesel sınırları olabilir ve çok yaklaştıklarında bir çarpışma algılanır.



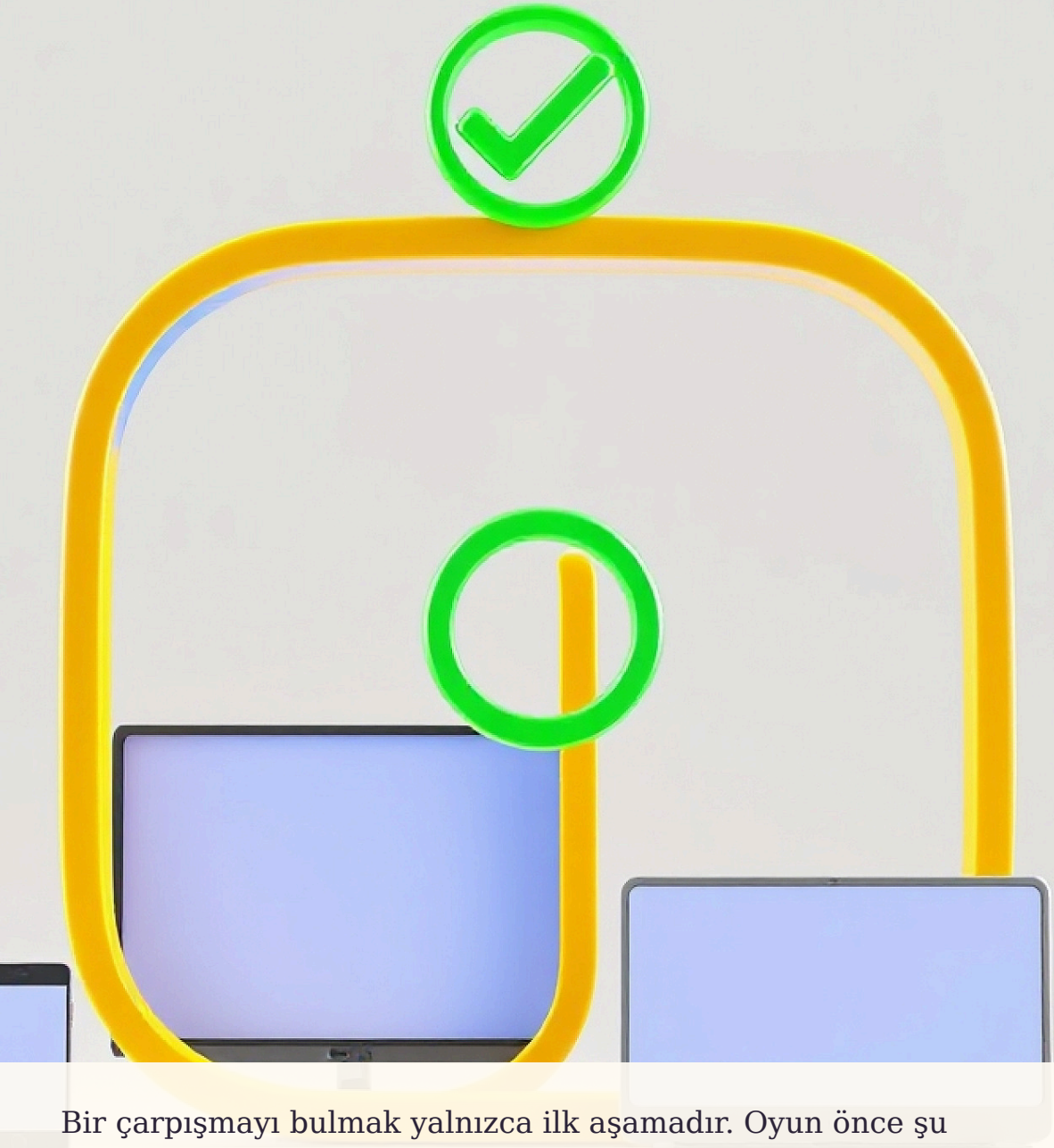
Bir duvara doğru hareket eden bir karakter düşün. Karakterin dikdörtgen bir sınırı, duvarın da dikdörtgen bir sınırı vardır. Karakter hareket ettikçe oyun, karakterin sınırının duvarın sınırıyla üst üste gelip gelmediğini kontrol eder. Üst üste geldiklerinde oyun bir çarpışma algılar. Örneğin, karakterin dikdörtgeni duvarın dikdörtgenine dokunursa oyun karakterin duvara çarptığını bilir.



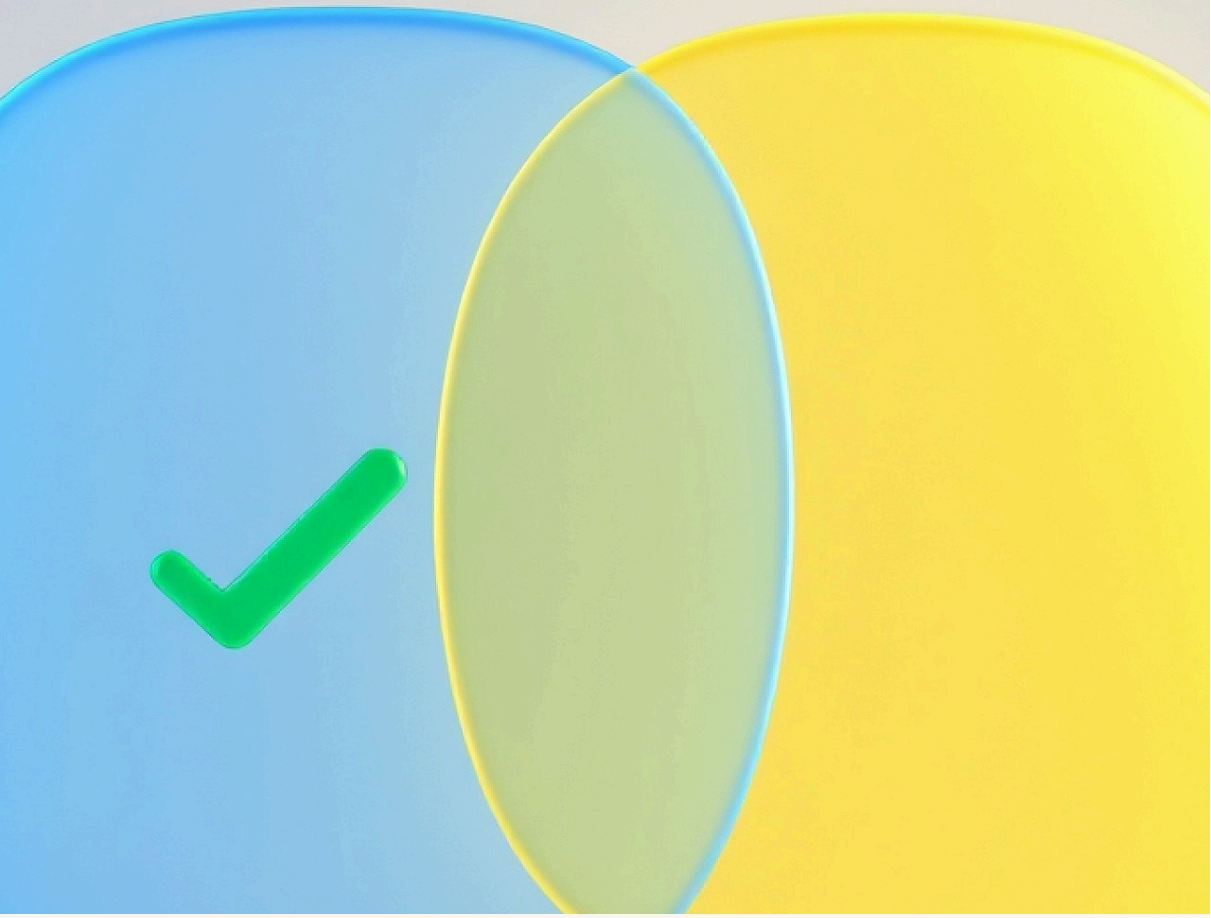
Bir nesnenin çevresindeki dikdörtgen veya daire, onun tam dış hattıyla örtüşmeyebilir. Örneğin, karmaşık şekilli bir karakter, bir miktar boş alan içeren bir dikdörtgenle yaklaşık olarak temsil edilebilir. Bu, basit oyunlar için sorun değildir, çünkü hızlıdır ve yeterince iyidir. Oyun, kusursuz doğruluk olmadan da çarpışmaları güvenilir şekilde algılayabilir.



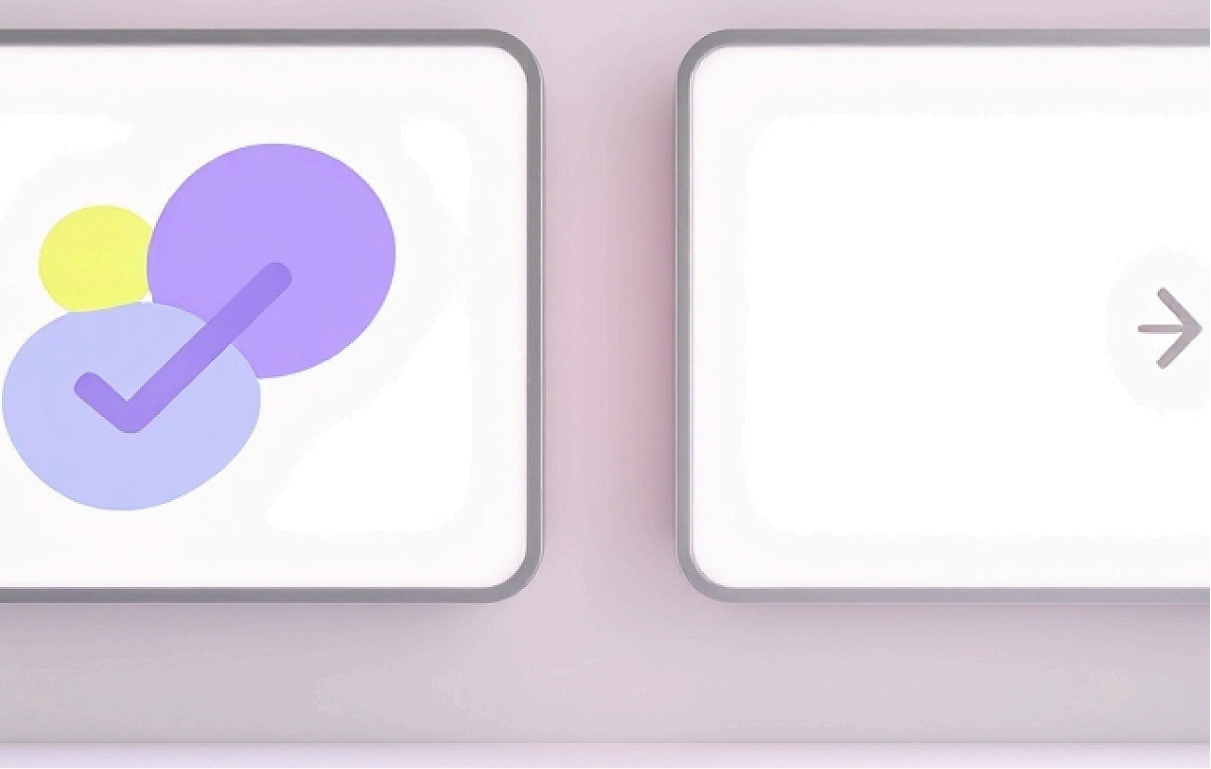
Daha ayrıntılı oyunlar, sınır olarak basit dikdörtgenler veya daireler yerine karmaşık şekiller kullanabilir. Bu şekiller, nesnenin dış hattını daha yakından izleyen çokgenler olabilir. Bu, bir nesnenin gerçek dış hattına daha yakın bir eşleşme sağlar. Örneğin, bir yarış oyunu, pist duvarlarıyla hassas çarpışmaları algılamak için bir araba için çokgen bir sınır kullanabilir.



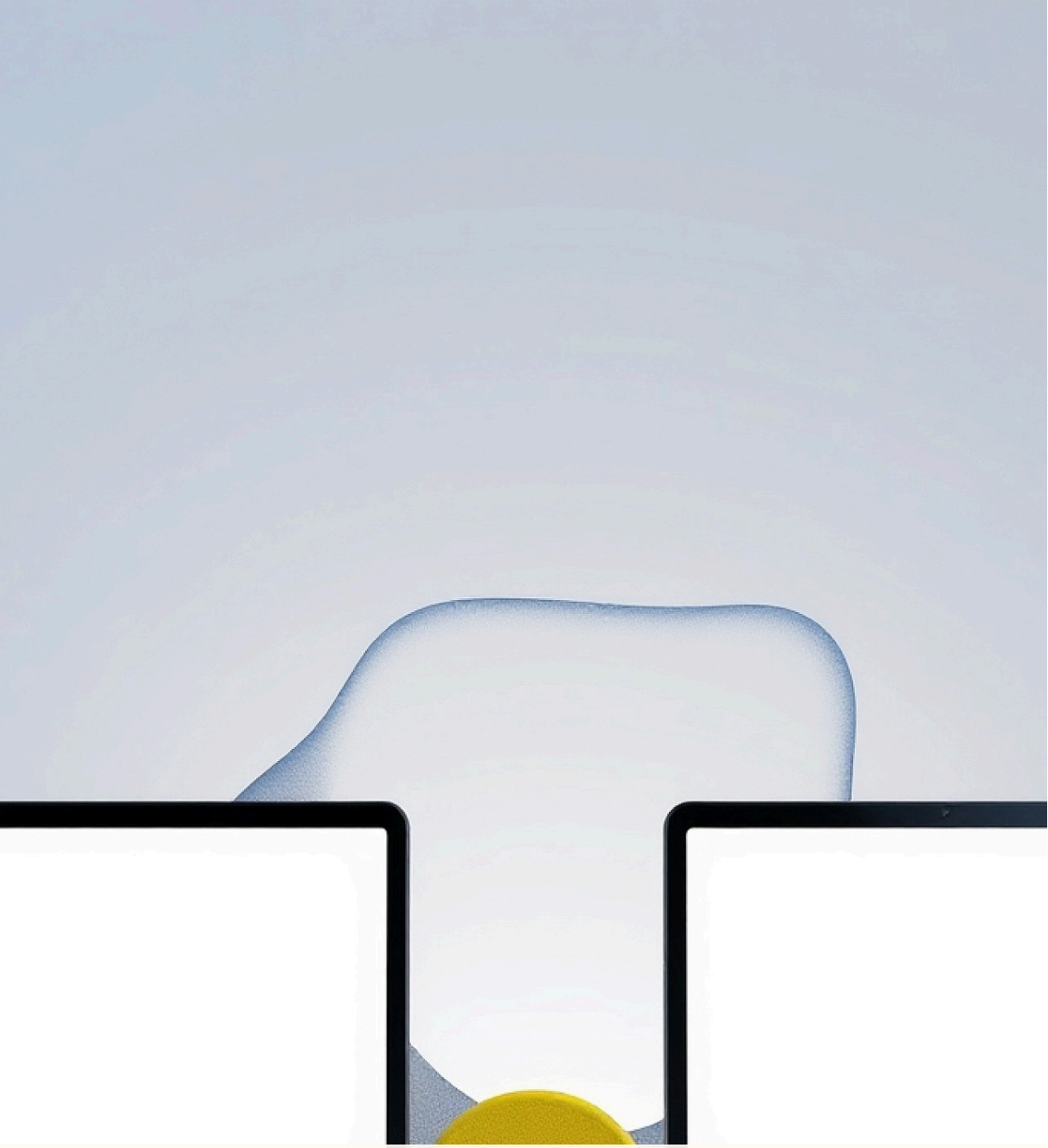
Bir çarpışmayı bulmak yalnızca ilk aşamadır. Oyun önce şu soruyu yanıtlar: sınırlar üst üste geldi mi? Bu aşamaya çarpışma algılama denir. Yalnızca üst üste gelmeyi kontrol eder ve sonrasında ne olacağına karar vermez. Örneğin, oyun bir karakterin sınırının bir duvarın sınırıyla üst üste geldiğini algılayabilir.



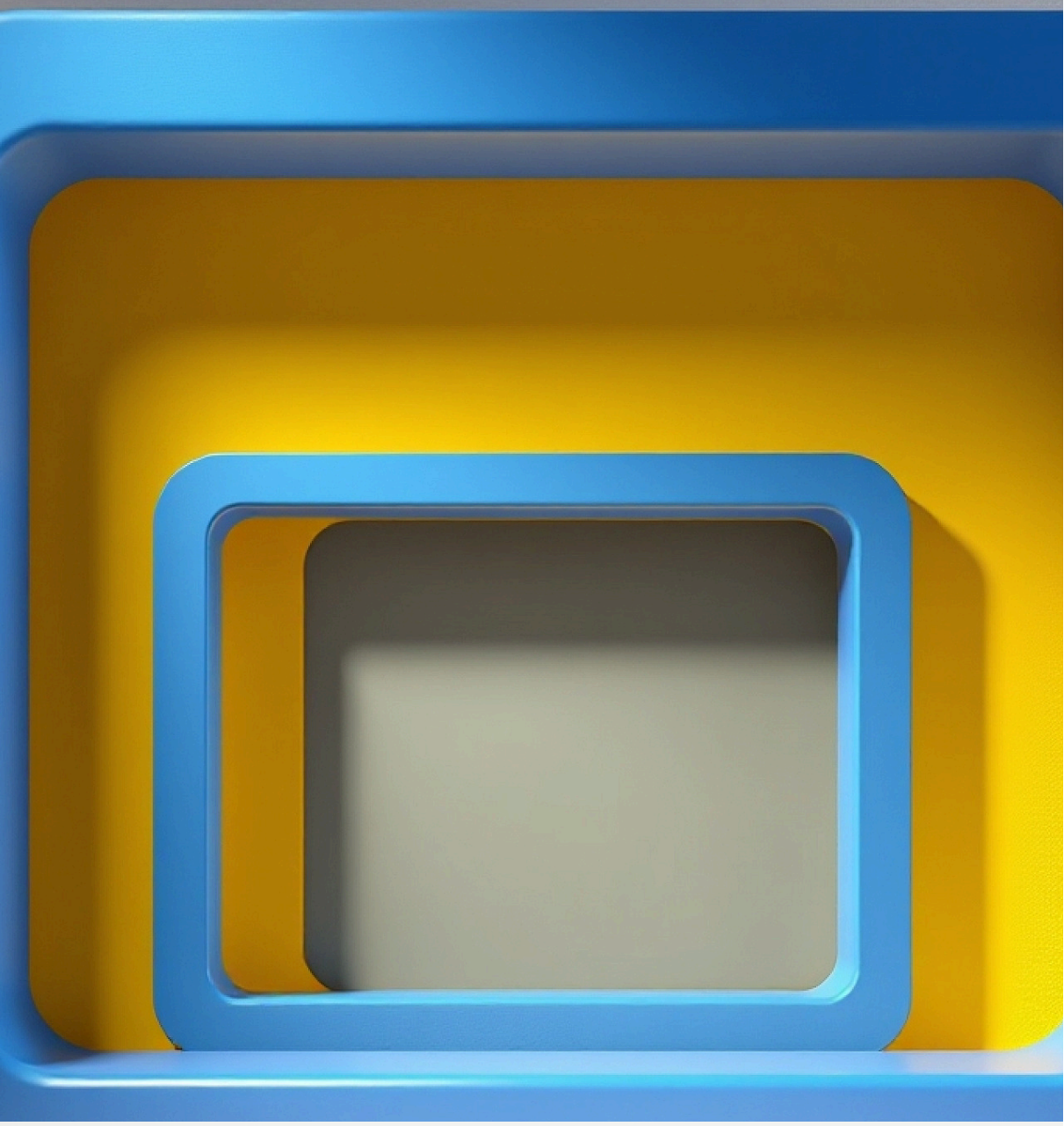
Nesnelerin bir çarpışmadan sonra nasıl tepki vereceğine karar vermek farklı bir aşamadır. Bir çarpışmayı algıladıktan sonra oyun karakteri durdurabilir, geri sektirebilir veya canını azaltabilir. Bu aşamaya çarpışma tepkisi denir. Dokunmayı algılamak ile tepkiyi hesaplamak ayrı tutulur. Örneğin, bir duvarla çarpışmayı algıladıktan sonra oyun karakteri önceki konumuna geri taşıyabilir.



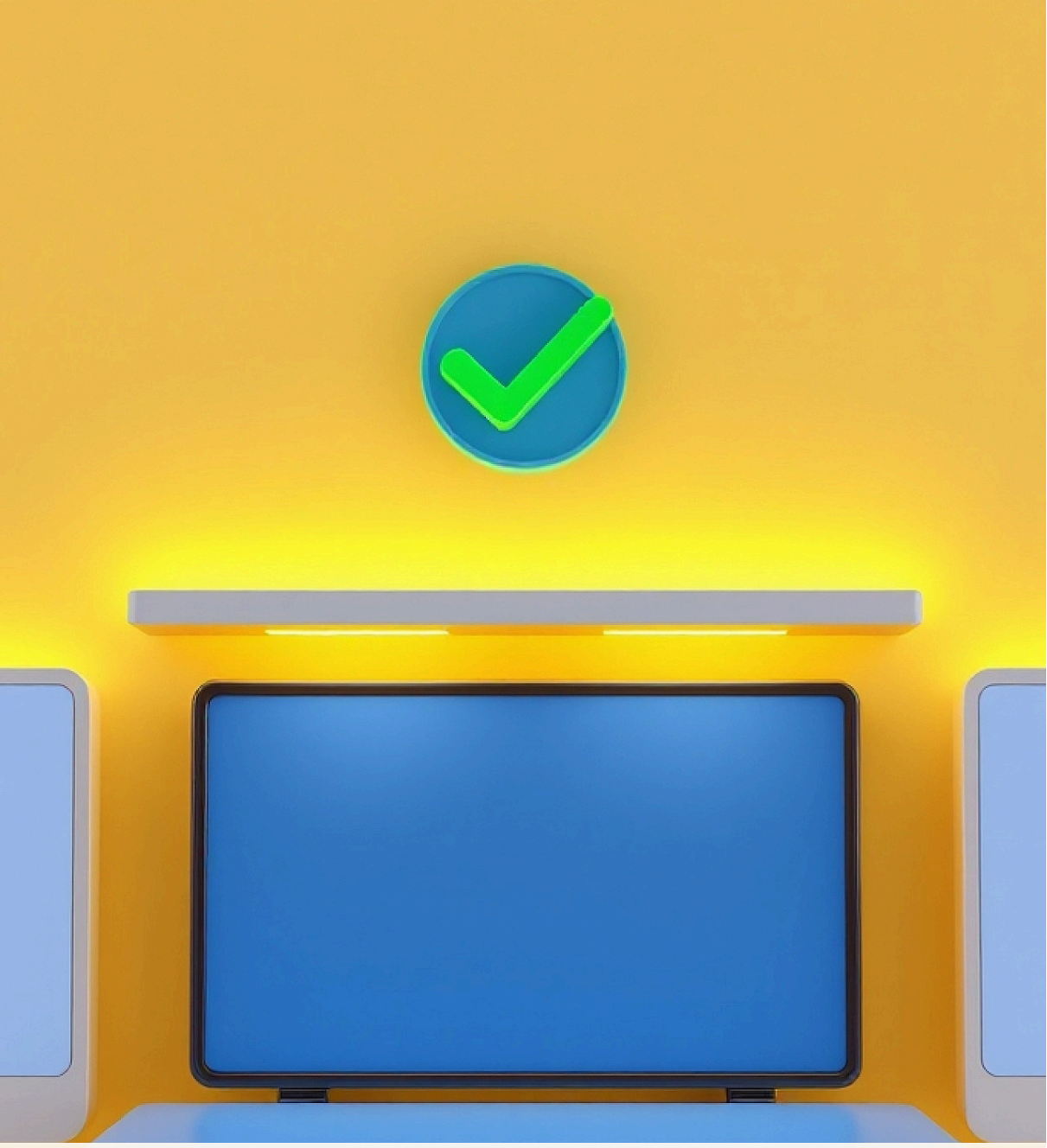
Yaygın bir hata, algılama ile tepkinin aynı şey olduğunu düşünmektir. Bunlar iki ayrı adımdır ve oyun bunları birbiri ardına ele alır. Önce oyun sınırları karşılaştırarak çarpışmayı algılar. Sonra tepkiyi hesaplar. Örneğin, oyun önce bir karakterin duvara çarptığını algılayabilir, sonra karakterin hareketini durdurabilir.



Bir başka hata, oyunun her zaman her nesnenin tam dış hattını kontrol ettiğini varsaymaktır. Basit oyunlar genellikle bunun yerine dikdörtgenler veya daireler gibi kaba sınırlar kullanır. Bunların hesaplanması daha hızlıdır ve birçok durum için yeterince iyidir. Örneğin, basit bir platform oyunu, görüntüleri daha ayrıntılı olsa bile tüm nesneler için dikdörtgenler kullanabilir.



Kaba bir sınır, bir dokunmayı biraz erken veya geç bildirebilir. Örneğin, bir karakterin çevresindeki dikdörtgen bir sınır, karakterin görünen kenarı bir duvara gerçekten dokunmadan önce bir çarpışma algılayabilir. Bu, basit çarpışma algılamada bilinen bir ödünleşimdir, bir hata değildir. Oyunu hızlı ve basit tutar.



Oyunlar, üst üste gelmeyi algılamak için nesne konumlarını ve sınırlarını karşılaştırır. Basit oyunlar genellikle dikdörtgenler veya daireler kullanırken ayrıntılı oyunlar karmaşık şekiller kullanabilir. Bir çarpışmayı algılamak ile tepkiyi hesaplamak ayrı aşamalardır. Örneğin, bir oyun önce bir karakterin sınırının bir duvarın sınırıyla üst üste gelip gelmediğini kontrol eder, sonra karakteri durdurmaya karar verir.