

Oyun döngüsü, FPS ve frameRate

Hareket ne kadar hızlı? Her bilgisayarda aynı hızda mı?

Güncelle ve çiz

FPS nedir?

frameRate()

Hz ile FPS farkı



SORU

Bir oyun, siz hiçbir tuşa basmasanız da neden durmaz?

Düşmanlar yürür, süre azalır, bulutlar kayar. Bunları kim, ne sıklıkla ilerletiyor?



Oyun döngüsü

Oyun döngüsü (game loop)

Oyun çalıştığı sürece tekrarlanan güncelleme döngüsüdür. Tipik adımları girdi okuma, mantık güncelleme ve çizimdir.

Girdiyi oku → Durumu güncelle → Ekranı çiz → Girdiyi oku → ...

- 1 `setup()` : tuval, başlangıç değişkenleri, bir kerelik ayarlar.
- 2 `draw()` : her frame'de arka plan, konumların güncellenmesi, çizim.
- 3 p5.js, sketch durdurulana kadar 2. adımı tekrarlar.

Döngüyü ağır çekimde izleyelim



Fareyi soldaki koyu alanda gezdirin. Top kırmızı çizgiye hemen gitmez: önce girdi okunur, sonra konum güncellenir, en son çizilir. Gerçek oyunda bu üç adım saniyede 60 kez döner.

Her frame 3 piksel sağa giden top

sketch.js

Kopyala

Düzenle

Büyüt

```
1 let topX = 50;
2 let topY = 200;
3 const TOP_YARICAP = 25;
4
5 function setup() {
6   createCanvas(500, 300);
7   noStroke();
8 }
9
10 function draw() {
11   background(30, 35, 50);
12
13   topX = topX + 3; // Her frame 3 piksel sağa
14
15   if (topX > width + TOP_YARICAP) {
16     topX = -TOP_YARICAP;
17   }
18
19   fill(255, 200, 80);
20   ellipse(topX, topY, TOP_YARICAP * 2, TOP_YARICAP * 2);
21 }
```

● Çıktı

Çalıştır

Büyüt



Önce konum güncelleniyor, sonra top çiziliyor. Sağdan çıkınca soldan yeniden giriyor.

SORU

Top görünmüyor. Ya da yanlış hızda gidiyor. Nereye bakarsınız?

`draw` 'ın içinde güncelleme ile çizim birbirine karışmışsa hatayı aramak zorlaşır.



Güncelle ve çiz: iki ayrı fonksiyon

sketch.js

Kopyala

Büyüt

```
1 function draw() {  
2   background(30, 35, 50);  
3   guncelleTop();  
4   cizTop();  
5 }  
6  
7 function guncelleTop() {  
8   topX = topX + 3;  
9   if (topX > width + TOP_YARICAP) {  
10    topX = -TOP_YARICAP;  
11  }  
12 }  
13  
14 function cizTop() {  
15   fill(255, 200, 80);  
16   ellipse(topX, topY, TOP_YARICAP * 2, TOP_YARICAP * 2);  
17 }
```

Top görünmüyorsa

cizTop 'a bakın.

Yanlış hızda gidiyorsa

guncelleTop 'a bakın.

Küçük örneklerde ikisi aynı fonksiyonda yazılabilir. Proje büyüdükçe ayırmak okumayı ve hata ayıklamayı kolaylaştırır.

setup ile draw ayrımı performanstır

setup içinde, bir kez

`createCanvas()` , görsel yükleme, büyük veri hazırlama gibi maliyetli işler.

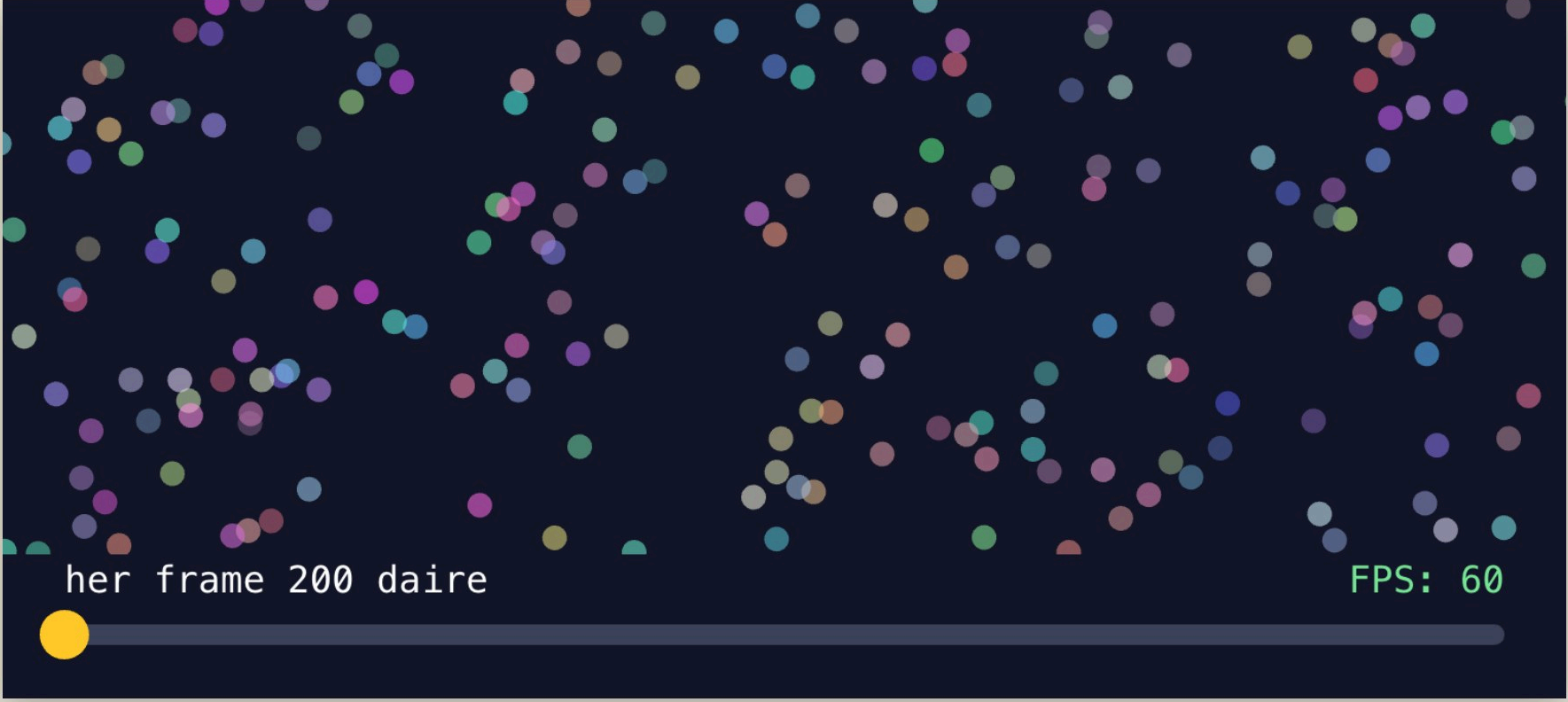
Yanlışlıkla draw içinde

Aynı iş her frame tekrarlanır ve FPS düşer.

Sketch yavaş çalışıyorsa ilk bakılacak yer: `draw` 'ın içinde her frame yapılması gerekmeyen bir iş var mı?

Bir frame'e ne kadar iş sığar?

ÇıktıÇalıştırBüyüt



her frame 200 daire

FPS: 60

Fareyi alttaki çubukta sağa götürün. `draw` içindeki iş arttıkça bir frame'in hazırlanması uzar ve FPS düşer. Sınır bilgisayardan bilgisayara değişir; sizinki kaçta düşmeye başlıyor?

FPS: saniyedeki frame sayısı

FPS (frames per second)

Bir saniyede `draw()` fonksiyonunun kaç kez çalıştığının ölçüsüdür.

Buradaki “frame”, geometrik bir kare değil; `draw()` çağrısının tek bir tekrarıdır.

- p5.js varsayılan olarak 60 FPS civarını hedefler.
- Donanım ve sketch'in yükü buna ulaşmayı engelleyebilir.
- Hedef, `frameRate()` ile ayarlanır.

Aynı hız, farklı FPS



Üç top da saniyede aynı yolu alıyor; yalnızca ekrana kaç kez çizildikleri farklı. En alttaki top “ışınlanıyor” gibi görünür: hızlı nesneleri gözle izlemek zorlaşır.

frameRate() : iki farklı kullanım

İfade	Anlamı	Sonuç
frameRate(60)	Hedef 60 FPS	Çoğu monitörde akıcı hareket
frameRate(30)	Hedef 30 FPS	Daha düşük yük, daha seyrek güncelleme
frameRate()	Ölçülen anlık FPS	Ekranda gösterilebilir (örneğin 58.2)

Hedef, garanti değildir

Sayı vererek çağırınca yalnızca hedefi belirlersiniz. Gerçek FPS donanıma ve çizim maliyetine bağlıdır; mobil ya da zayıf cihazlarda 60 her zaman tutmaz.

FPS'i ekranda görelim

sketch.js

Kopyala

Düzenle

Büyüt

```
1 function setup() {  
2   createCanvas(400, 200);  
3   frameRate(60);  
4 }  
5  
6 function draw() {  
7   background(40);  
8   fill(255);  
9   textSize(18);  
10  textAlign(LEFT, TOP);  
11  text("Ölçülen FPS: " + nf(frameRate(), 0, 1), 15, 15);  
12 }
```

● Çıktı

Çalıştır

Büyüt

Ölçülen FPS: 59.9

"Düzenle" ile `frameRate(60)` değerini 10, sonra 5 yapın. Ölçülen değer nasıl değişiyor?

`nf()` : sayıyı okunur yazmak

Parametre	Görevi
<code>sayi</code>	Biçimlenecek değer
<code>solBasamak</code>	Noktanın solundaki basamak sayısı; gerekirse başa 0 ekler
<code>sagBasamak</code>	Noktanın sağındaki basamak sayısı; yuvarlar

Çağrı	Sonuç
<code>nf(58.273, 0, 1)</code>	<code>"58.3"</code>
<code>nf(58.273, 0, 0)</code>	<code>"58"</code>
<code>nf(7, 3, 0)</code>	<code>"007"</code>
<code>nf(3.14159, 1, 2)</code>	<code>"3.14"</code>

`nf(sayi, solBasamak, sagBasamak)` yalnızca gösterimi düzenler; FPS hesabını değiştirmez.

SORU

$\text{topX} = \text{topX} + 3$. FPS 60'tan 30'a düşerse top ne yapar?

Bir saniyede kaç piksel yol alıyordu? Şimdi kaç piksel alır?



Yarı hızda gider

sketch.js

Kopyala

Düzenle

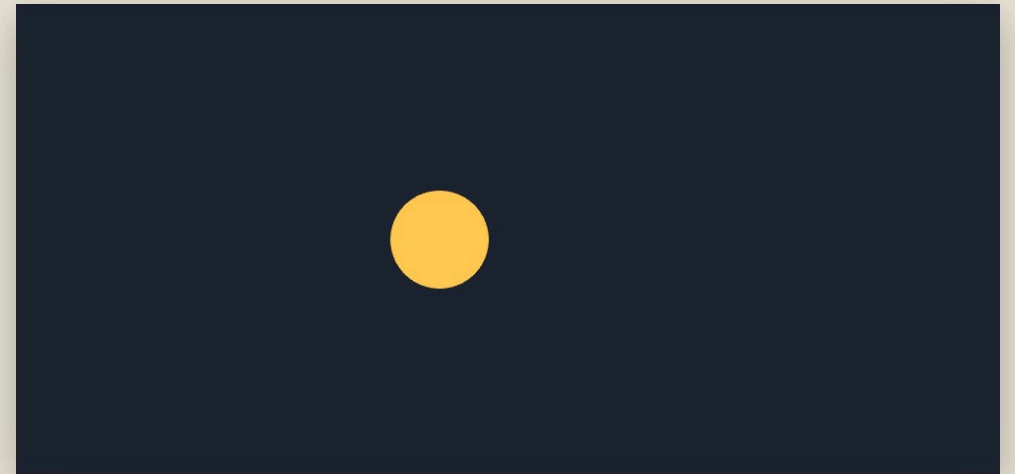
Büyüt

```
1 let topX = 50;
2 const TOP_YARICAP = 25;
3
4 function setup() {
5   createCanvas(500, 240);
6   noStroke();
7   frameRate(30); // 60 yapıp yeniden çalıştırın
8 }
9
10 function draw() {
11   background(30, 35, 50);
12
13   topX = topX + 3; // her frame 3 piksel
14   if (topX > width + TOP_YARICAP) {
15     topX = -TOP_YARICAP;
16   }
17
18   fill(255, 200, 80);
19   ellipse(topX, 120, TOP_YARICAP * 2, TOP_YARICAP * 2);
20 }
```

Çıktı

Çalıştır

Büyüt



Hız, frame sayısına bağlı

60 FPS

Saniyede $60 \times 3 = 180$ piksel.

30 FPS

Saniyede $30 \times 3 = 90$ piksel.

Sonuç

Aynı kod, farklı cihazlarda farklı hızda oynar.

Çözüm, hızı frame sayısına değil **geçen gerçek süreye** bağlamak: `deltaTime` . Onu bir sonraki konuda göreceğiz.

Monitörün hızı ile oyunun hızı aynı şey değil

Kavram	Birim	Ne ölçer?	Tipik değerler
Yenileme hızı	Hz	Monitörün saniyede kaç kez yeni görüntü çizdiği	60, 75, 120, 144, 165, 240...
FPS	frame/s	Oyunun saniyede kaç frame ürettiği	p5.js'te hedef 30–60; oyunlarda 60–240+

Oyun frame üretir → Monitör o frame'leri (ya da bir kısmını) gösterir

Çoğu ofis ve dizüstü ekranı 60 Hz'dir. Oyun monitörlerinde 120–144 Hz yaygın; telefon ve tabletlerde 90 / 120 Hz paneller sıklaşıyor.

Tarayıcı ikisini eşlemeye çalışır

Vsync benzeri davranış

Tarayıcılar çizimi monitörün yenileme döngüsüyle eşlemeye çalışır. Yırtılma (tearing) azalır. Oyun, panelin gösterebildiğinden fazla frame üretirse fazlası atılır ya da birleştirilir.

Yüksek Hz'li monitörde

`frameRate()` çağrılmazsa p5.js daha yüksek FPS'e çıkmaya çalışabilir. Frame başına sabit piksel ekleyen hareket o zaman daha hızlı oynar.

`frameRate(60)` yazıp hedefi açıkça vermek, paneller arası farkı azaltır. Asıl güvenilir çözüm yine `deltaTime`.

Sinema 24, oyun 60: neden?

24

SinemaKlasik film uzun süre yaklaşık 24 fps ile özdeşleşti. Sessiz dönemde 16–18 fps de yaygındı.

25–30

TelevizyonBölgeye göre yaklaşık 25 ya da 30 fps'lik yayın aileleri.

30–120+

OyunEtkileşim daha yüksek hız ister: 30 oynanabilir, 60 akıcı; rekabetçi oyunlarda 120 ve üstü istenir.

“Kaç fps iyi?” sorusunun cevabı ortama göre değişir. Film izlemek ile nişancı oyunu oynamak aynı ihtiyaç değil.

Oyunlarda FPS neden önemli?

Akıcılık

Düşük ve dalgalı FPS hareketi kesik kesik gösterir; kalabalık sahnelerde rahatsız eder.

Tepki hissi

Daha sık çizilen frame, tıklamanın sonucunu ekrana daha erken yansıtır.

Adalet

Hareket frame'e bağlıysa yüksek FPS'li oyuncu avantaj ya da dezavantaj yaşar.

Okunabilirlik

Hızlı nesnelerde (mermi, top) seyrek frame "nereye gitti?" hissi verir.

Ünlü oyunlar kaç FPS'te çalışır?

Zelda: Ocarina of Time 1998

Saniyede yaklaşık 20 frame.
Dönemin donanımı üç boyutlu bir dünyayı ancak bu hızda çizebiliyordu; oyun yine de klasik oldu.

Doom 1993

Oyunun mantığı saniyede 35 adım atar. Hızlı hissi veren, yüksek FPS değil, tutarlı tempodur.

E-spor bugün

Rekabetçi nişancı oyunlarında 144, 240 hatta 360 Hz monitörler kullanılır; oyun da o kadar frame üretmeye çalışır.

Patlama sahnesinde FPS 9'a düştü.



oyuncu

Bu artık oyun değil, slayt gösterisi.



yorumlar

Daha yüksek her zaman daha iyi mi?

Algı

“Göz X fps’nin üstünü göremez” diye tek bir sihirli sayı yok. 30 → 60 geçişi belirgin bir yumuşama olarak hissedilir; 60 → 144 daha çok hızlı kamera ve nişan almada fark eder. Yükseldikçe kazanç azalır.

Donanım

Daha fazla frame, işlemciye ve ekran kartına daha çok iş demektir: güç tüketimi ve ısı artar, pil çabuk biter, fanlar hızlanır. Oyunlar bu yüzden FPS sınırı ve pil modu sunar.

Asıl önemli olan istikrar

Sürekli 55–60, bir anda 90’a çıkıp 40’a düşen bir grafikten daha öngörülebilir bir oynanış verir.

Yapay zekâyla: FPS düşerse

 10-15 dk

1. `x += 3` ile giden bir kare düşünün. FPS 60'tan 30'a inince ne olur? Önce kendiniz yazın.
2. Aynı senaryoyu bir yapay zekâya sorun; gerekçesini sizinkiyle karşılaştırın.
3. Sonraki bölümde göreceğimiz `deltaTime`'in bu sorunu nasıl çözeceğini tek cümleyle tahmin edin.

Her mini oyun için dört soru

1

VeriDurumu hangi
değişkenler tutuyor?

2

GüncellemeHer
frame'de ya da olayda
ne değişiyor?

3

ÇizimEkkranda
görünenler hangi
değişkenlerden
geliyor?

4

KuralOyuncunun
eylemi hangi koşulla
kabul ya da
reddediliyor?

Eksik kalan adımı (bir sınır, bir başlangıç değeri, tek seferlik bir olay) şimdi tamamlamak, sonra büyük bir sketch'in içinde aramaktan kolaydır.

Tek cümlelik köprü 5 dk

“Şimdi **X biliyorum; bundan sonra **Y** için buna ihtiyacım var.”**

Bu cümleyi bugünkü konu için doldurun. Kuramıyorsanız, eksik kalan başlığın örneğini bir kez daha çalıştırın.

Kod zamanı

 30 dk

1. Kayan top örneğine ölçülen FPS'i yazdırın. Topun tuvali baştan sona kaç saniyede geçtiğini ölçün.
2. `frameRate(30)` yapın ve yeniden ölçün. Tahmininiz tuttu mu?
3. Kodu `guncelleTop()` ve `cizTop()` olarak ikiye bölün. İkinci bir top ekleyin: daha yavaş ve başka renkte.
4. Bilerek yavaşlatın: `draw` içinde 20 000 turluk bir `for` ile küçük daireler çizin. FPS kaçta düştü?
5. **Zorlu:** top kenardan çıkıp başa dönmek yerine iki kenar arasında gidip gelsin; her sekmede rengi değişsin.

60 FPS'te top 500 piksellik tuvali 5 saniyede geçiyor. Kod $x += 2$. FPS 20'ye düşerse?

A Yine 5 saniye

B 10 saniye

☒ **C** Yaklaşık 15 saniye

D Yaklaşık 1,7 saniye

Frame sayısı üçte bire indi; saniyede alınan yol da üçte bire iner. Süre üç katına çıkar.

Hangisi `draw` içinde olmamalı?

A `background(30)`

B `topX = topX + 3`

☒ `createCanvas(500, 300)`

D `ellipse(topX, 150, 50, 50)`

Tuval bir kez kurulur. `draw` 'a yazılırsa saniyede 60 kez yeniden kurulur ve FPS düşer.

`frameRate(120)` yazdınız. Ekranda ölçülen FPS 60
görünüyor. Neden olabilir?

A

`frameRate` yalnızca 60'a kadar sayı kabul eder.



Sayı yalnızca bir hedeftir; monitörün yenileme hızı ya da donanım daha fazlasına izin vermiyor olabilir.

C

`nf` sayıyı yarıya böler.

Tarayıcı çizimi monitörün yenileme döngüsüyle eşler. 60 Hz bir ekranda 60'ın üstüne çıkmak çoğu zaman mümkün olmaz.

Merak edenler için kaynaklar

ROBERT NYSTROM · GAME PROGRAMMING PATTERNS

Game Loop

Oyun döngüsünün neden ve nasıl kurulduğunu anlatan, ücretsiz okunabilen kitap bölümü (İngilizce).

gameprogrammingpatterns.com/game-loop.html

BLUR BUSTERS

UFO Test

Aynı hareketi farklı FPS'lerde yan yana gösteren test sayfası; kendi monitörünüzü deneyin.

testufo.com

P5.JS REFERANSI

frameRate()

Hedef FPS'i ayarlamak ve ölçülen değeri okumak.

p5js.org/reference/p5/frameRate

GLENN FIEDLER · GAFFER ON GAMES

Fix Your Timestep!

İleri düzey: profesyonel oyunlarda güncelleme ile çizimin ayrılması (İngilizce).

gafferongames.com/post/fix_your_timestep

Bölüm 2'yi bitirdik

- ✓ Oyun döngüsü: girdiyi oku, durumu güncelle, ekranı çiz; sonra yeniden.
- ✓ Güncelleme ile çizimi ayırmak hata ayıklamayı kolaylaştırır.
- ✓ Maliyetli işler `setup` 'a aittir; `draw` 'a yazılırsa FPS düşer.
- ✓ FPS: saniyede `draw` 'ın çalışma sayısı. `frameRate(60)` hedef verir; `frameRate()` ölçer.
- ✓ Hz monitörün, FPS oyunun hızıdır.
- ✓ `x += 3` gibi frame'e bağlı hareket, her cihazda farklı hızda oynar.

BÖLÜM 2 BİTTİ

Sırada: diziler ve deltaTime

Birçok nesneyi tek bir listede tutmak ve her cihazda aynı hızda hareket etmek.

[Tüm sunumlar](#)[← Önceki](#)