Ders 2

Ön Tasarım

Önemli Bir Hata

- Pek çok projede oyun tasarım süreçlerinden bir an önce kodlamaya geçilmek istenir

Fikir -> Konsept -> Tasarım -> Kod

- Halbuki oyunlar dışındaki yazılım projelerinde böyle değildir
- Tasarım genelden özele yayılmalıdır

80/20 Kuralı

- Asla tasarımın %100 bitmesini beklemeyin
- %80 tasarım bittiğinde kodlamaya başlayın
- Kalan %20 zaman içinde şekillenir

Tasarım - Kod Dengesi

- Kodlamaya başladığınızda mutlaka tasarım dökümanınızı da paralel olarak geliştirin
- Kod tasarımı, tasarım kodu yansıtsın
- Kod tasarım dökümanı değildir, onun yerine geçmez

Game Loop Tasarımı

Nedir?

- Game loop, bir oyunun zamana bağlı akışını sağlayan kod bloğudur
- Belirli bir sırayla oyunun tüm elemanlarını çağırır / aktive eder
- Loop'un her dönüşü, bir frame'e denk gelir
  - Oyun mekaniği hesapları: matematik, fizik, AI
  - Çizim: Render

Eski Zaman Problemleri

- İlk başlarda game loop artarda çalıştırılırdı
- Bu dönemlerde cihazlarda çeşitlilik çok azdı
- Tüm bilgisayarlar aşağı yukarı aynı özellikteydi
- Her frame kıymetliydi, başka işlere ayıracak zaman yoktu

Sonrası ...

- El konsollorının populer olması ile birlikte, sabit FPS yaygınlaştı
- Bu yöntemde oyun her zaman maksimum bir FPS oranında çalışır
- El konsolları için bu iyidir
- Ama PC için büyük problemdir!
  - PC ne kadar iyi olursa olsun, oyun kalitesi değişmez
  - PC zayıfsa, oyun yavaşlar!

Semi-decoupling

- Çizim ve mekanik ayrı alt-looplara yüklenir
- Mekanik sabit FPS ile döner
- Çizim bağımsız ve sınırsız döner
- Çizim yavaş kalırsa, frame atlamış olur
- Mekanik yavaş kalırsa, aynı frame iki defa çizilir (veya pas geçilir)
- SORUN: hala makinenizin tüm gücü kullanılmıyor

Full-decoupling

- Mekanik sınırsız hızda çalıştırılır
- Her iki frame arası geçen zaman ölçülür -> delta
- Tüm mekanik delta üzerinden hesaplanır

```
currentTime = System.getTimeInMillis();
elapsedTime = lastTime - currentTime;
lastTime = currentTime;
delta = 1.0 / elapsedTime;

Px += Vx * delta;
```

Dikkat

- Mekanik işlemler yarıda iken çizim yapılırsa ne olur?
- Nasıl çözülebilir?

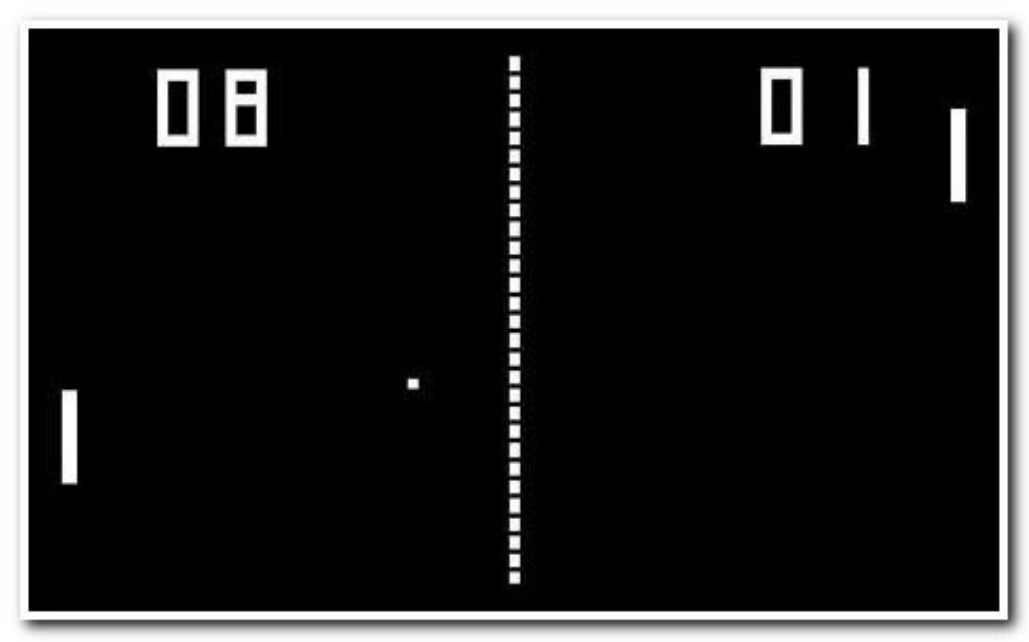
Oyun nedir?

Oyunların ortak noktaları nelerdir?

Oyunların ortak noktaları nelerdir?

- Tüm oyunlarda en az bir oyuncu vardır
- Oyuncu yoksa o bir animasyondur!
- Tüm oyunlarda **token**lar vardır
  - Oyuncunun dolaylı veya direkt olarak etkilediği nesneler
  - Bu tokenlar oyun mekaniği çerçevesinde bilgisayar tarafından yönetilir

Pong



- Oyuncular (sopa)
- Top
- Skorlar
- Alt ve üst duvar
- Kaleler

Token hiyerarşisi

- Oyun dünyası
  - Oyuncu
    - Sopa, Skor
  - Duvar
    - Alt, Üst
  - Kale
    - Sağ, Sol
  - Top

Etkileşim matrisi

	Sopa	Top	Duvar	Kale	Skor
Sopa	X				
Top	Çarpışma (yansı)	X			
Duvar	Çarpışma (dur)	Çarpışma (yansı)	X		
Kale	X	Çarpışma (tetikle:Gol)		X	
Skor	X	X	X	Gol (arttır)	X

Etkileşim matrisi

- Yorucu görünebilir, ama uzun vadede hayatı kolaylaştırır
- Birşeyleri gözden kaçırmanızı engeller
  - A ile B çarpışırsa ne olacak?

Oyun dünyası token'ı

- Oyun dünyası karmaşıklaştıkça, oyun dünyasının kendisi için de bir token tutmak gerekir
- Bu token aynı zamanda diğer tokenlar arasındaki iletişimi de sağlar
  - Token-token
    - Her token diğer tüm tokenları tanır
  - token-world-token
    - Oyun dünyası tokenlar arasında arabuluculuk yapar

Yeni bir token eklemek

- Hangi sistemde daha kolay olur?

Gelecek hafta için görevler

- 5 adet oyun motorunun tanıtımını yapınız
  - Hangi dilleri kullanıyor?
  - Hangi platformlarda çalışıyor?
  - Hangi platformlar için oyun üretiyor?
  - 2D/3D/2.5D
  - Özellik eklentileri: multiplayer, AI, modelleme, vs.
  - Yapılan oyunlardan örnekler
  - Lisans ve ücretlendirme