



BURSA ULUDAĞ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
2024 – 2025 Eğitim Öğretim Yılı Bahar Dönem
Bilgisayar Grafikleri
Ray Tracing Final Raporu

Ad Soyad - Numara

Mehmet Halim Baş - 032290157

Murat Berk Yetiştirir - 032290008

1. Grup Üyelerinin Görev Dağılımı

- **Murat Berk Yetiştirir:**

- Teorik araştırma ve literatür taraması yaparak, DirectX12 Ray tracing mimarisi, API detayları ve uygulama senaryolarını derinlemesine incelemek.
- Uygulamaların çalıştırılması, görsellerin hazırlanması.
- Rapor ve sunum metnlerinin oluşturulmasına katkıda bulunmak.

- **Mehmet Halim Baş:**

- Uygulamaların analiz edilmesi, karşılaştırmalı raporlanması.
- Çalışma sırasında elde edilen verilerin düzenlenmesi.
- Teorik araştırma ve literatür taraması yapılması.
- Rapor ve sunum metnlerinin oluşturulmasına katkıda bulunmak

2. Proje Adımları

- **Araştırma ve Literatür Taraması:**

- DirectX12 Ray tracing teknolojisinin teknik dökümantasyonlarının incelenmesi (Microsoft DirectX-Graphics-Samples, NVIDIA DXR örnekleri).
- Makaleler YouTube videoları üzerinden uygulama örneklerinin analiz edilmesi.

- **Uygulamaların İncelenmesi:**

- Microsoft'un resmi DirectX-Graphics-Samples deposu ve DXR Minimal API Sample'ın incelenmesi ve çalışma prensiplerinin rapor edilmesi.
- NVIDIA'nın DXR örnek projelerinin çalıştırılması, uygulamaların denenmesi ve elde edilen sonuçların ekran görüntüleriyle belgelenmesi.

- **Analiz ve Rapor Hazırlama:**

- Elde edilen verilerin ve uygulama sonuçlarının analiz edilmesi, karşılaştırmalı görseller ile desteklenmesi.

- **Sonuç ve Değerlendirme:**

- Ray tracing teknolojisinin avantajları, karşılaşılan zorluklar ve gelecekteki geliştirme potansiyelinin tartışılması.

3. Projenin Amacı ve Kapsamı

Bu projenin amacı, Microsoft'un DirectX 12 API'si ile birlikte gelen Ray Tracing (ışın izleme) teknolojisinin temel prensiplerini açıklamak, DXR (DirectX Raytracing) bileşenlerini tanıtmak ve bu teknolojinin gerçek zamanlı grafiklerde nasıl uygulandığını örneklerle göstermektir. Projede hem teorik bilgiler hem de demo uygulamalar aracılığıyla ışığın fiziksel olarak doğru simülasyonunun oyun ve grafik motorlarına nasıl entegre edilebileceği incelenmiştir.

Projenin kapsamı şunları içermektedir:

- DirectX 12 API'sinin grafik programlamadaki rolünün açıklanması
- Ray Tracing'in geleneksel rasterization yöntemleriyle farklarının vurgulanması
- DXR teknolojisinin temel bileşenlerinin (Acceleration Structures, Shader Binding Table, Shader türleri) detaylandırılması
- NVIDIA RTX kartları ile donanımsal hızlandırmanın nasıl gerçekleştiğinin aktarılması
- Ray tracing'in performans üzerindeki etkisinin ve kalite-performans dengesinin analiz edilmesi
- Demo sahneler aracılığıyla yansımalar, gölge ve ışık kırılması gibi efektlerin gerçek zamanlı uygulanmasının gösterilmesi

Bu kapsamda, hem teknik altyapıya dair bilgi sunulmakta hem de geliştiricilere yönelik gerçekçi grafik sahneleri oluşturmak için rehber niteliğinde örnekler sunulmaktadır.

4. Demo Çıktısı

- Demo videosu drive linki: [RayTracingKayıt.mp4](#)
- Demo ekran görüntüleri: [RayTracingResimler](#)

5. Demo Kodları

- Demo kodları linki: [D3D12Raytracing](#)
- Demo dosyaları linki: [Bilgisayar Grafikleri 2025](#)

6.DirectX 12 Nedir?

DirectX 12, Microsoft tarafından geliştirilen bir grafik ve multimedya API'sidir. Windows işletim sistemlerinde ve Xbox platformlarında yüksek performanslı oyunlar ve grafik uygulamaları geliştirmek için kullanılır.

DirectX, oyun motorları ve grafik uygulamaları ile ekran kartı (GPU) arasında bir köprü görevi görür. DirectX'in sunduğu bu arayüz sayesinde geliştiriciler, donanıma erişim sağlayarak 3B grafikler, animasyonlar ve efektler oluşturabilirler.

6.1 DirectX 12'nin Amacı

- GPU ve CPU'yu daha verimli kullanmak.
- Donanıma daha düşük seviyede erişim sağlamak.
- Büyük ve karmaşık oyunlarda daha yüksek performans sunmak.
- Çok çekirdekli işlemcilerden maksimum fayda sağlamak.

7. Ray Tracing Nedir?

Ray tracing (ışın izleme), ışığın gerçek dünyadaki gibi davranmasını taklit eden bir tekniktir. Işık, nesnelere çarpınca yansır, gölge yapar ya da kırılır. Ray tracing bunları hesaba katarak çok gerçekçi görüntüler oluşturur. Normal oyunlardaki grafiklerden daha canlı ve sinematik durur.

7.1. Gerçek Dünya Işık Modellemesi

Ray tracing, bilgisayar grafiklerinde ışığı gerçek dünyadaki gibi taklit etmeye çalışan bir tekniktir. Yani ışık bir kaynaktan çıkar, nesnelere çarpar, yansır, gölgeler oluşturur veya cam gibi yüzeylerde kırılır. Tıpkı gözümüzle gördüğümüz gibi ışığın davranışını hesaplayarak çok doğal ve etkileyici görüntüler üretir.

7.2. Geleneksel Rasterization ile Farkları

Rasterization (geleneksel yöntem), sahnedeki nesneleri üçgenlere bölerek doğrudan ekrana çizer. Ancak bu yöntem ışığın sahneyle olan detaylı etkileşimini basitleştirir. Örneğin gölgeler veya yansımalar ayrı ayrı kodlanmak zorundadır. Ray tracing ise ışığı gerçekten izlediği yolu üzerinden hesapladığı için bu tür efektleri otomatik ve daha doğru bir şekilde verir.

7.3. Fiziksel Gerçekçiliği Nasıl Sağlar?

Ray tracing, her bir piksele bir ışın gönderir ve bu ışının sahnede nereye çarptığını, nelerden yansıdığını veya geçip geçmediğini hesaplar. Bu sayede yansımalar, gölgeler, ışık kırılması gibi fizik kurallarına dayalı efektler oluşturur. Bu da görüntüyü neredeyse bir fotoğraf kadar gerçekçi yapar. Özellikle metal, cam gibi yüzeylerde fark çok belirgindir.

8. DirectX 12 ile Ray Tracing (DXR)

DirectX Raytracing (DXR), Microsoft'un DirectX 12 API'si içinde sunduğu bir uzantıdır. DXR sayesinde geliştiriciler, fiziksel olarak doğru ışık simülasyonları (yansımalar, gölgeler, kırılmalar) uygulayabilir. Bu da sahnelerin çok daha gerçekçi ve sinematik görünmesini sağlar. Ray tracing önceden sadece sinema endüstrisinde kullanılırken, artık DX12 sayesinde gerçek zamanlı oyunlarda da kullanılabilir.

8.1 DXR Bileşenleri Nelerdir?

Ray tracing, geleneksel rasterization'dan farklı olarak sahnedeki ışınların fiziksel hesaplarını yapar. Bunu yapmak için DXR, aşağıdaki bileşenleri kullanır:

8.1.1 Acceleration Structures (İvme Yapıları)

- Ray'lerin sahnedeki nesnelerle hızlıca kesişip kesişmediğini kontrol eder.
- Hiyerarşik yapılardır:
 - **Bottom-Level Acceleration Structure (BLAS)**: Nesne düzeyinde
 - **Top-Level Acceleration Structure (TLAS)**: Sahne düzeyinde
- Amaç: Işının hangi nesneyle çarpıştığını minimum hesapla bulmak.

8.1.2 Shader Binding Table (SBT)

- Farklı ışın türlerine (raygen, miss, hit vb.) hangi shader'ların uygulanacağını belirtir.
- Bir tür "shader haritası"dır.
- GPU'nun hangi ışın için hangi işlemi yapacağını tanımlar.

8.1.3. Shader Türleri

Shader Türü	Görevi
Ray Generation Shader	Işınları başlatır (kamera gibi)
Miss Shader	Işın hiçbir şeyle kesişmezse ne yapılacağını belirler (arka plan vs.)
Closest Hit Shader	Işının çarptığı en yakın yüzeyde ne yapılacağını hesaplar
Any Hit Shader	Işın geçerken ara kontroller yapar (örneğin saydamlık)
Intersection Shader	Gelişmiş şekillerle kesişme kontrolü sağlar (özelleştirme amaçlı)

8.2 RTX Kartlar ile Donanımsal Destek

NVIDIA RTX serisi ekran kartları, ray tracing işlemleri için özel RT Core adı verilen donanım birimlerine sahiptir.

Bu çekirdekler:

- Işın-üçgen kesişimlerini donanım seviyesinde hızlandırır.
- CPU ve standart GPU işlemlerine göre kat kat hızlı çalışır.
- DXR ile doğrudan uyumludur.

8.3 Performans ve Kalite Dengesi

Ray tracing, son derece güçlü ve gerçekçi görseller sunan bir teknolojidir; ancak hesaplama maliyeti oldukça yüksektir. Bu nedenle geliştiriciler genellikle rasterizasyon ve ray tracing'in birleştiği "Hybrid Rendering" yöntemini tercih eder.

DLSS gibi yapay zekâ destekli teknolojilerle birlikte kullanıldığında, görüntü kalitesi büyük ölçüde korunurken kare hızı (FPS) yüksek tutulabilir. Bu sayede RTX ve DLSS teknolojileri bir araya geldiğinde, hem yüksek görsel kalite hem de oynanabilir performans elde edilir. DX12 Raytracing (DXR), Acceleration Structures, Shader Binding Table, Raygen, Miss ve Hit shader'ları gibi bileşenleri içerir ve bu özellikler RTX ekran kartları tarafından desteklenir. Böylece performans ve kalite arasında dengeli bir çözüm sunulmuş olur.

9. Avantajlar ve Dezavantajlar

9.1 Avantajlar

Ray tracing, grafiklerde ışığı çok daha gerçekçi işler. Işığın nesnelerden yansması, camdan kırılması, doğru yerde ve açıda gölgeler oluşturması gibi detaylar doğal görünür. Bu sayede sahneler adeta bir film karesi gibi olur. Özellikle su, cam, metal gibi yüzeylerdeki yansımalar çok etkileyicidir.

9.2 Dezavantajlar

Bu kadar gerçekçilik, bilgisayara daha fazla yük bindirir. Ray tracing çalıştırmak için güçlü bir ekran kartı gerekir. Ayrıca sistem daha çok hesaplama yaptığı için oyunlarda FPS düşebilir. Yani görüntü kalitesi artsa da akıcılık azalabilir. Ayrıca kodlama kısmı daha karmaşıktır, bu yüzden herkes kolayca kullanamaz.

9.3 Kullanım Alanları

- **Oyunlar:** Özellikle yeni nesil oyunlarda gerçekçi atmosfer için kullanılır.
- **Sinema:** Animasyon filmleri ve görsel efektlerde çoktan standart hale gelmiştir.
- **Mimarlık ve tasarım:** Gerçekçi ışık ve gölge ile binaların, iç mekânların nasıl görüneceği önceden simüle edilebilir.

10. Sonuç ve Değerlendirme

Proje kapsamında DirectX 12 ve Ray Tracing teknolojileri detaylı şekilde araştırılmış, teknik dökümantasyonlar ve örnek uygulamalar incelenerek kapsamlı bir literatür taraması gerçekleştirilmiştir. Microsoft'un resmi DXR demo projeleri ile çalıştırılmış, elde edilen çıktı ve görseller analiz edilerek raporlanmıştır.

DirectX Ray Tracing (DXR) bileşenlerinin sahne oluşturma sürecine etkisi; yansıma, gölge ve ışık kırılması gibi unsurlar üzerinden değerlendirilmiştir. Donanımsal destek (RTX kartlar) ile birlikte gerçek zamanlı Ray Tracing'in performansa etkisi gözlemlenmiş; hem görsel kalite hem de FPS dengesi açısından elde edilen veriler karşılaştırmalı olarak sunulmuştur.

Proje, hem teorik hem de pratik düzeyde DXR teknolojisinin grafik motorlarında nasıl kullanılabileceğini göstermiştir. Elde edilen sonuçlar, Ray Tracing'in gerçek zamanlı uygulamalarda sunduğu yüksek görsellik avantajını açıkça ortaya koymaktadır. Ancak yüksek donanım gereksinimi ve karmaşık kod yapısı gibi zorluklar da dikkat çekicidir.

11. KAYNAKÇA

- Microsoft DirectX-Graphics-Samples [GitHub Repository]:
(<https://github.com/microsoft/DirectX-Graphics-Samples>)
- Wikipedia: ([https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)))
- DXR Minimal API Sample – Microsoft Developer Documentation:
(<https://docs.microsoft.com/en-us/windows/desktop/direct3d12/d3d12-raytracing>)
- NVIDIA DXR Sample – NVIDIA GameWorks
(<https://github.com/NVIDIAGameWorks/DXR-OptiX>)
- Real-Time Ray Tracing - NVIDIA:
(https://developer.nvidia.com/rtx/ray-tracing?sortBy=developer_learning_library%2Fsort%2Ftitle%3Aasc)
- YouTube videoları
 - <https://www.youtube.com/watch?v=ezXGTRSx1g8>
 - <https://www.youtube.com/watch?v=hUVfAVjsfL4>
- Makaleler
 - https://link.springer.com/content/pdf/10.1007/978-1-4842-4427-2_25.pdf
 - <https://arxiv.org/abs/2312.06827>
- ChatGPT