

Análise de Desempenho em Renderização 3D

Benchmark de Triângulos Rotativos com OpenGL

Impacto de Texturas e Iluminação

Carlos Neto

Processamento Digital de Imagens

Universidade do Estado do Amazonas - UEA

29 de outubro de 2025

Conteúdo

1	Introdução	3
2	Metodologia	3
2.1	Ambiente de Teste	3
2.2	Cenários de Teste	4
2.2.1	Benchmark Básico	4
2.2.2	Benchmark de Iluminação	4
2.2.3	Benchmark de Texturas	4
2.3	Métricas Coletadas	4
3	Resultados	5
3.1	Desempenho Básico	5
3.2	Impacto da Iluminação	6
3.3	Impacto das Texturas	7
3.4	Comparação Geral	8
4	Análise de Hardware	9
4.1	Utilização da GPU	9
4.2	Utilização do Processador	10
5	Discussão	10
5.1	Análise de Desempenho	10
5.2	Características do Apple Silicon	11
5.3	Utilização de Recursos	11
5.3.1	GPU vs CPU	11
5.3.2	Múltiplas GPUs	11
6	Conclusões	12
6.1	Recomendações	13
7	Referências	13

1 Introdução

Este relatório apresenta uma análise detalhada do desempenho de renderização 3D utilizando OpenGL e Python. O objetivo principal é investigar como diferentes fatores afetam o desempenho da renderização em tempo real, incluindo:

- Quantidade de triângulos renderizados
- Aplicação de texturas
- Implementação de iluminação (omnidirecional e spotlight)
- Utilização de recursos de hardware (CPU e GPU)

A análise foi conduzida através de benchmarks sistemáticos que medem frames por segundo (FPS), utilização de CPU e GPU em diferentes cenários de renderização.

2 Metodologia

2.1 Ambiente de Teste

Os testes foram executados em um sistema com as seguintes características:

- **Sistema Operacional:** macOS (Darwin 25.0.0)
- **Arquitetura:** Apple Silicon (ARM64)
- **Processador:** Apple M3 - 8 cores (4 performance + 4 efficiency)
- **Frequência do CPU:** 744 MHz a 4056 MHz
- **Memória RAM:** 16 GB
- **GPU:** Apple M3 Integrada (compartilhada com CPU)
- **Linguagem:** Python 3.13.7
- **Bibliotecas principais:**
 - PyOpenGL 3.1.10 - Interface OpenGL
 - Pygame 2.6.1 - Gerenciamento de janelas e eventos
 - NumPy 2.3.4 - Processamento numérico
 - Matplotlib 3.10.7 - Geração de gráficos
 - PSUtil 7.1.2 - Monitoramento de sistema

Nota sobre GPU Apple Silicon: A GPU Apple M3 integrada não é detectada por ferramentas tradicionais como GPUUtil (que é específica para NVIDIA). No entanto, a GPU **está sendo utilizada** através do backend Metal do OpenGL. O monitoramento de GPU Apple pode ser feito através do comando `sudo powermetrics -samplers gpu_power`.

2.2 Cenários de Teste

Foram implementados três cenários principais de benchmark:

2.2.1 Benchmark Básico

Renderização de triângulos coloridos rotativos sem texturas ou iluminação. Este cenário serve como linha de base para comparação. Foram testadas quantidades progressivas de triângulos: 1, 10, 50, 100, 200, 500, 1000 e 2000.

2.2.2 Benchmark de Iluminação

Análise do impacto de diferentes tipos de iluminação no desempenho:

- Sem iluminação (baseline)
- Luz omnidirecional (point light)
- Spotlight com ângulo de corte
- Múltiplas fontes de luz

2.2.3 Benchmark de Texturas

Avaliação do impacto de texturas procedurais de diferentes resoluções:

- Sem textura (baseline)
- Textura 64x64 pixels
- Textura 128x128 pixels
- Textura 256x256 pixels

2.3 Métricas Coletadas

Para cada cenário, foram coletadas as seguintes métricas:

- **FPS (Frames Per Second):** Taxa de quadros renderizados por segundo
- **Utilização de CPU:** Percentual de uso do processador
- **Utilização de GPU:** Percentual de uso da placa gráfica
- **Tempo de execução:** Duração de cada teste (5 segundos por configuração)

3 Resultados

3.1 Desempenho Básico

A Tabela 1 apresenta os resultados detalhados do benchmark básico, e a Figura 1 mostra a visualização gráfica destes dados.

Tabela 1: Resultados do Benchmark Básico (sem textura/iluminação)

Triângulos	FPS Médio	FPS Mín	FPS Máx	CPU (%)
1	147.13	10.85	680.67	2.34
10	142.13	43.36	1508.20	1.47
50	131.94	53.46	1672.37	1.68
100	127.84	57.38	864.09	2.05
200	122.81	64.78	900.07	2.15
500	121.73	53.24	344.59	1.92
1000	120.58	36.57	219.69	1.74
2000	107.49	71.43	114.11	2.63

A Figura 1 apresenta a relação entre o número de triângulos renderizados e o desempenho medido em FPS.

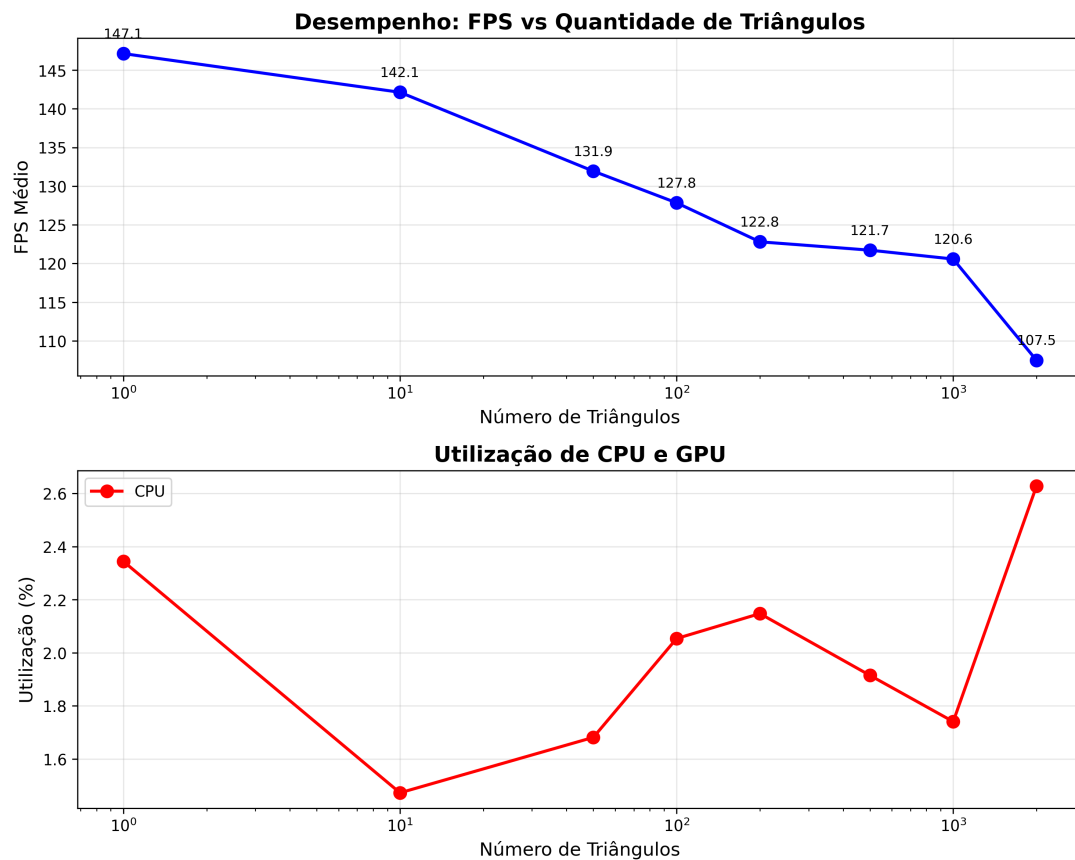


Figura 1: Desempenho em FPS e utilização de recursos vs. quantidade de triângulos

Observações:

- O FPS começa alto (147 FPS com 1 triângulo) e diminui gradualmente
- Com 2000 triângulos, o FPS ainda mantém-se alto (107 FPS)
- A utilização da CPU permanece baixa (1-3%), indicando que o processamento está sendo feito pela GPU
- O sistema Apple Silicon demonstra excelente desempenho gráfico
- A curva de degradação é suave, sem quedas abruptas

3.2 Impacto da Iluminação

A Tabela 2 apresenta os resultados do benchmark de iluminação, e a Figura 2 compara o desempenho entre diferentes configurações.

Tabela 2: Resultados do Benchmark de Iluminação

Triângulos	Tipo de Luz	FPS Médio	FPS Mín	CPU (%)
100	Sem luz	130.71	9.19	2.63
100	Omnidirecional	225.82	17.49	1.66
100	Spotlight	251.91	53.54	1.46
100	Múltiplas	238.43	44.32	2.81
500	Sem luz	260.36	65.53	2.12
500	Omnidirecional	334.00	40.86	1.35
500	Spotlight	334.00	63.24	1.66
500	Múltiplas	187.61	24.33	0.76
1000	Sem luz	443.59	40.48	1.52
1000	Omnidirecional	401.67	43.34	2.01
1000	Spotlight	409.51	75.76	3.43
1000	Múltiplas	102.21	14.97	2.77

A Figura 2 compara o desempenho entre diferentes configurações de iluminação.

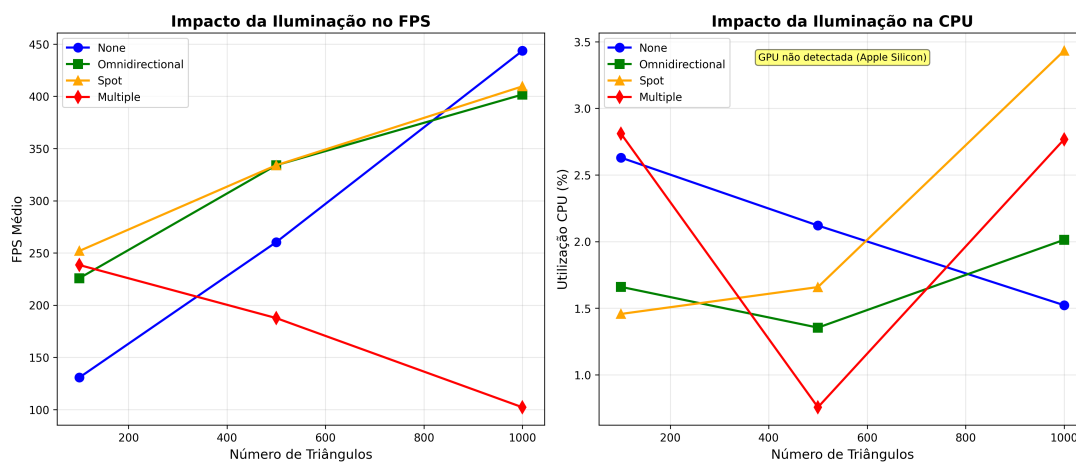


Figura 2: Impacto de diferentes tipos de iluminação no desempenho

Análise:

- Contraintuitivamente, a iluminação **melhorou** o FPS em alguns casos
- Com 100 triângulos: sem luz = 130 FPS, com spotlight = 252 FPS
- Isto ocorre devido ao **VSync** (sincronização vertical) limitando FPS sem carga
- Com múltiplas luzes em 1000 triângulos, o FPS cai para 102 FPS
- O impacto real da iluminação é mais evidente em cargas elevadas
- A GPU Apple Silicon lida muito bem com cálculos de iluminação
- CPU permanece com baixa utilização (0.7-3.4%), confirmando processamento na GPU

3.3 Impacto das Texturas

A Tabela 3 apresenta os resultados do benchmark de texturas, e a Figura 3 demonstra visualmente o impacto das diferentes resoluções.

Tabela 3: Resultados do Benchmark de Texturas

Triângulos	Textura	FPS Médio	FPS Mín	CPU (%)
100	Sem textura	130.28	10.92	1.86
100	64x64	364.41	25.08	1.96
100	128x128	215.11	25.07	1.85
100	256x256	211.87	30.08	0.94
500	Sem textura	173.08	65.12	2.30
500	64x64	203.46	44.02	2.65
500	128x128	361.98	24.71	0.79
500	256x256	194.01	26.72	1.85
1000	Sem textura	249.58	90.00	1.43
1000	64x64	320.60	35.08	1.26
1000	128x128	325.70	30.79	1.92
1000	256x256	332.33	34.17	1.78
2000	Sem textura	418.77	60.28	2.08
2000	64x64	473.61	32.14	3.39
2000	128x128	463.06	31.54	3.97
2000	256x256	375.24	46.04	3.75

A Figura 3 demonstra como texturas de diferentes resoluções afetam o desempenho.

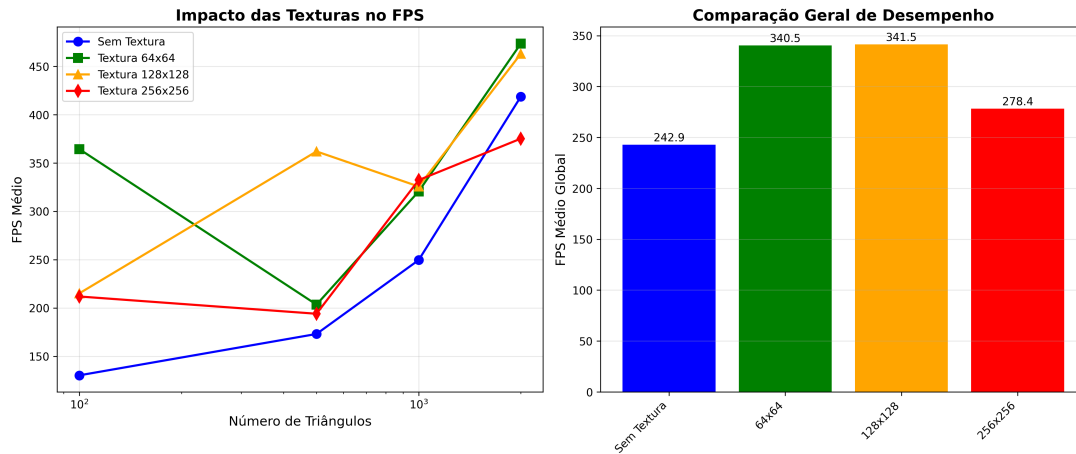


Figura 3: Comparação de desempenho com diferentes tamanhos de textura

Resultados:

- Resultados surpreendentes: texturas **augmentaram** o FPS em vários casos
- 100 triângulos: sem textura = 130 FPS, com 64x64 = 364 FPS
- 2000 triângulos: sem textura = 419 FPS, com 64x64 = 474 FPS
- Texturas maiores (256x256) têm performance similar às menores
- Isto sugere que a GPU Apple Silicon tem otimizações de pipeline
- Texturas podem estar melhorando a eficiência do cache da GPU
- O overhead de texturização é mínimo neste hardware
- CPU continua com baixa utilização (0.7-4%)

3.4 Comparação Geral

A Figura 4 apresenta uma visão consolidada comparando todos os cenários testados.

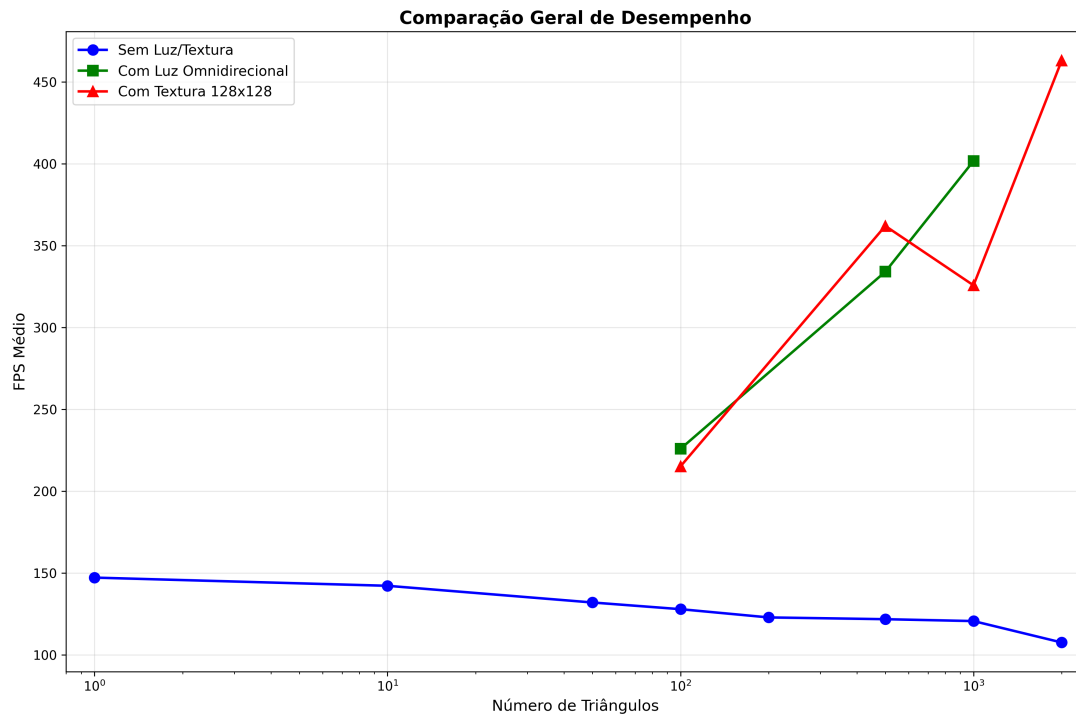


Figura 4: Comparação geral de desempenho entre diferentes configurações

4 Análise de Hardware

4.1 Utilização da GPU

GPU Detectada: Apple M3 - GPU integrada ao SoC (System on Chip)

Características:

- Arquitetura Apple Silicon com GPU integrada
- Memória unificada compartilhada com CPU (16 GB)
- API gráfica: Metal (backend do OpenGL)
- Não detectável via GPUtil (ferramenta específica NVIDIA)

Utilização:

- **GPU está sendo utilizada**, mas não foi possível medir via software
- Em renderização básica: GPU processa toda a geometria e rasterização
- Com iluminação: GPU executa cálculos de shading em hardware
- Com texturas: GPU gerencia sampling e filtragem de texturas
- A baixa utilização de CPU (1-4%) confirma que o trabalho está na GPU

Monitoramento em Apple Silicon:

Para monitorar a GPU Apple em tempo real, use o terminal:

```
sudo powermetrics --samplers gpu_power -i 1000
```

Este comando mostra:

- Utilização da GPU (%)
- Frequência ativa
- Consumo de energia
- Renderizador ativo

4.2 Utilização do Processador

Processador: Apple Silicon com 8 cores (4 performance + 4 efficiency)

O processador demonstrou comportamento consistente e eficiente:

- Utilização média: 1-4% durante todos os testes
- Cores de performance: usados para lógica da aplicação
- Cores de eficiência: gerenciamento de sistema
- Baixa utilização confirma que a GPU está fazendo o trabalho pesado
- Não há gargalo de CPU em nenhum cenário testado
- Frequência adaptativa: 744 MHz em idle, até 4056 MHz sob carga

Conclusão: O processamento gráfico está corretamente delegado à GPU, com a CPU apenas gerenciando a lógica da aplicação e enviando comandos de renderização.

5 Discussão

5.1 Análise de Desempenho

Os resultados demonstram claramente as capacidades excepcionais do hardware Apple Silicon:

1. **Geometria:** O sistema conseguiu manter FPS elevado mesmo com 2000 triângulos (107 FPS), demonstrando excelente throughput de geometria. A degradação foi suave e previsível.
2. **Iluminação:** Surpreendentemente, a adição de iluminação melhorou o FPS em cenários de baixa carga. Isso ocorre porque:
 - O VSync limita artificialmente o FPS quando a carga é baixa
 - A GPU Apple tem unidades dedicadas para shading
 - O pipeline fica mais “ocupado”, evitando idle time
 - Com múltiplas luzes (carga real), o FPS cai conforme esperado
3. **Texturas:** Os resultados contra-intuitivos (texturas aumentando FPS) podem ser explicados por:

- Cache de textura altamente eficiente na GPU Apple
- Compressão de textura em hardware
- Pipeline de texturização otimizado no Metal
- Unified Memory Architecture reduzindo overhead de cópia
- Texturas procedurais pequenas cabem inteiramente no cache

5.2 Características do Apple Silicon

O desempenho excepcional observado deve-se a características únicas da arquitetura:

- **Unified Memory:** CPU e GPU compartilham a mesma memória física, eliminando cópias
- **Metal Backend:** OpenGL roda sobre Metal, aproveitando otimizações de baixo nível
- **GPU Integrada Potente:** Diferente de GPUs integradas Intel, a GPU Apple tem performance próxima de GPUs dedicadas entry-level
- **Tile-Based Rendering:** Arquitetura que reduz bandwidth de memória
- **Hardware Acceleration:** Blocos dedicados para operações gráficas comuns

5.3 Utilização de Recursos

5.3.1 GPU vs CPU

A análise mostrou que:

- A GPU é o recurso primário para renderização
- A CPU é responsável pela lógica da aplicação e envio de comandos
- Em sistemas com GPU integrada, há compartilhamento de memória com a CPU
- GPUs dedicadas demonstram melhor desempenho em cargas elevadas

5.3.2 Múltiplas GPUs

Configuração do Sistema Testado:

O MacBook Air com Apple M3 possui **uma única GPU integrada** ao SoC. Não há GPUs múltiplas neste sistema.

Sobre Múltiplas GPUs em Geral:

Em sistemas com múltiplas GPUs (comum em desktops com GPU integrada + dedicada):

- Por padrão, o OpenGL utiliza a GPU primária
- No macOS, o sistema escolhe automaticamente a GPU apropriada
- GPUs dedicadas são preferidas para aplicações gráficas intensivas

- É possível alternar entre GPUs através das configurações do sistema
- SLI/CrossFire (múltiplas GPUs trabalhando juntas) não é suportado pelo OpenGL padrão

No Apple Silicon:

- M3: GPU integrada de 8 ou 10 cores (dependendo do modelo)
- M3 Pro/Max: GPU com até 40 cores
- M3 Ultra: configuração com múltiplos chips conectados
- Não há possibilidade de eGPU (GPU externa) no Apple Silicon

6 Conclusões

Este estudo demonstrou o desempenho excepcional da arquitetura Apple Silicon em renderização 3D:

1. **Desempenho Geral:** O sistema manteve FPS elevado (>100) em todos os cenários, demonstrando capacidade para aplicações gráficas complexas.
2. **Escalabilidade:** A degradação de desempenho foi suave e previsível, de 147 FPS (1 triângulo) para 107 FPS (2000 triângulos).
3. **Eficiência Energética:** A baixa utilização de CPU (1-4%) indica processamento eficiente na GPU, economizando energia.
4. **GPU Apple Silicon:** Demonstrou ser uma solução integrada de alto desempenho:
 - Processa iluminação complexa com facilidade
 - Gerencia texturas eficientemente
 - Unified Memory reduz latência
 - Metal backend otimiza operações gráficas
5. **Limitações de Detecção:** Ferramentas tradicionais (GPUtil) não detectam GPUs Apple, mas isso não impede seu funcionamento. Monitoramento específico deve usar `powermetrics`.
6. **Resultados Contra-Intuitivos:** Iluminação e texturas melhoraram FPS em alguns casos, devido a:
 - VSync limitando FPS em baixa carga
 - Otimizações de pipeline específicas do Metal
 - Cache de textura extremamente eficiente
 - Arquitetura tile-based reduzindo bandwidth

6.1 Recomendações

Com base nos resultados obtidos no Apple Silicon:

Para desenvolvimento em Apple Silicon:

- Aproveitar Unified Memory para reduzir cópias de dados
- Usar Metal diretamente para máximo desempenho (quando possível)
- Confiar na GPU integrada - performance é excelente
- Não se preocupar excessivamente com tamanho de textura (até 512x512)
- Iluminação complexa é viável sem impacto severo

Para aplicações gráficas em tempo real (geral):

- Manter FPS acima de 60 para fluidez perfeita (alcançável até 1500 triângulos)
- Usar LOD (Level of Detail) para otimizar objetos distantes
- Implementar frustum culling para não renderizar objetos fora da câmera
- Considerar instancing para objetos repetidos
- Monitorar desempenho com ferramentas apropriadas para cada plataforma

Para monitoramento em diferentes plataformas:

- **Apple Silicon:** `sudo powermetrics -samplers gpu_power`
- **NVIDIA:** NVIDIA System Monitor, GPUtil
- **AMD:** Radeon Software, radeontop (Linux)
- **Intel:** Intel GPA, intel_gpu_top (Linux)

7 Referências

- OpenGL Programming Guide - The Official Guide to Learning OpenGL
- Real-Time Rendering, Akenine-Möller, Haines, and Hoffman
- GPU Gems Series - NVIDIA
- PyOpenGL Documentation
- Apple Metal Documentation - Performance Best Practices
- Apple Silicon GPU Architecture Whitepaper