

## Semana 14: Online - Teoria

João Pedro Bianchini Ramos - 32087411

Lucas Tonin Leite - 32089759

Victor Costa Nova Alexandre - 32052960

A diferença entre as operações `"tex1 * tex2"` e `"mul(tex1, tex2)"` está relacionada à forma como essas operações são interpretadas no contexto de uma linguagem de shader, como o HLSL (High-Level Shading Language).

### **Justificativa:**

Quando usamos o operador `"*"` entre duas texturas, como em `"tex1 * tex2"`, a interpretação é de que estamos realizando uma multiplicação de componentes de cor entre as duas texturas. Nesse caso, cada componente de cor das texturas (como vermelho, verde, azul e, possivelmente, alfa) é multiplicado separadamente, resultando em uma nova textura com os componentes resultantes da multiplicação.

Por outro lado, a função `"mul(tex1, tex2)"` é uma função de multiplicação fornecida pela linguagem de shader HLSL. Essa função realiza a multiplicação de forma semelhante ao operador `"*"`, mas pode ter algumas diferenças específicas dependendo da implementação do HLSL ou da arquitetura da GPU.

### **Diferenças**

Uma diferença possível entre as duas operações é que o operador `"*"` pode ser otimizado pelo compilador do shader para utilizar instruções específicas da GPU para multiplicação de componentes de cor, aproveitando os recursos de hardware disponíveis. Já a função `"mul()"` pode ter uma implementação mais genérica, sem otimizações específicas para multiplicação de texturas, o que pode resultar em um desempenho ligeiramente inferior.

Além disso, é importante ressaltar que a forma como as texturas são amostradas e filtradas também pode influenciar os resultados das operações. Dependendo das configurações de amostragem e filtragem de texturas aplicadas, os resultados podem ser diferentes mesmo ao realizar as mesmas operações.

## Conclusão

Em resumo, as diferenças nas operações "tex1 \* tex2" e "mul(tex1, tex2)" estão relacionadas às interpretações específicas da linguagem de shader e possíveis otimizações de hardware. É importante considerar as configurações de amostragem e filtragem de texturas, bem como as peculiaridades da implementação do shader e da arquitetura da GPU, para entender completamente as diferenças nos resultados.

## Referências

Documentação oficial do HLSL: A Microsoft fornece documentação detalhada sobre a linguagem de shading HLSL.

<https://learn.microsoft.com/en-us/windows/win32/direct3dhls/dx-graphics-hlsl>

Xuejun Li and Meng Xu, "Water simulation based on HLSL," *2009 IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China, 2009, pp. 1066-1069, doi: 10.1109/ICNIDC.2009.5360906. disponível em :

<https://ieeexplore.ieee.org/abstract/document/5360906> Acesso: 22 maio 2023

## Livros:

HAINES, Eric et al. **Real-time rendering**. Crc Press, 2018. disponível em :  
<https://books.google.com.br/books?hl=pt-BR&lr=&id=0g1mDwAAQBAJ&oi=fnd&pg=PP1&dq=real+time+rendering&ots=0k-HIFxEah&sig=cNwNbfZsBsCXVNsJzplIFo2iCNGo> Acesso: 22 maio 2023

AKENINE-MÖLLER, Tomas; HAINES, Eric; HOFFMAN, Naty. **Real-time rendering**. Crc Press, 2019. Disponível em:  
[https://books.google.com.br/books?hl=pt-BR&lr=&id=Y-KEDwAAQBAJ&oi=fnd&pg=PP1&dq=real+time+rendering&ots=o5V0eBJ\\_C7&sig=zPrAVzjnRh6rmoRSar26lbBc2Og](https://books.google.com.br/books?hl=pt-BR&lr=&id=Y-KEDwAAQBAJ&oi=fnd&pg=PP1&dq=real+time+rendering&ots=o5V0eBJ_C7&sig=zPrAVzjnRh6rmoRSar26lbBc2Og) Acesso: 22 maio 2023

:

