

GODOT
Meetup

São Paulo - SP

Tornando seu jogo mais cinematográfico com Color Correction LUT



GODOT
meetup

O que é Color Grading?

O que é Color Grading?

Técnica de pós-produção para
correção de cor.

A man with a beard and a dark beanie is shown from the chest up, looking slightly to his left. He is wearing a light-colored jacket with a dark fur collar. The image is split vertically down the middle. The left half is labeled 'BEFORE' and has a cool, blueish-grey color grade. The right half is labeled 'AFTER' and has a warmer, more orange and yellow color grade, suggesting a sunset or sunrise. The background consists of snow-covered hills under a cloudy sky.

BEFORE

AFTER







E Color Grading LUT?

Uma técnica de pós-produção para correção de cor em **aplicações em tempo real (games)**.

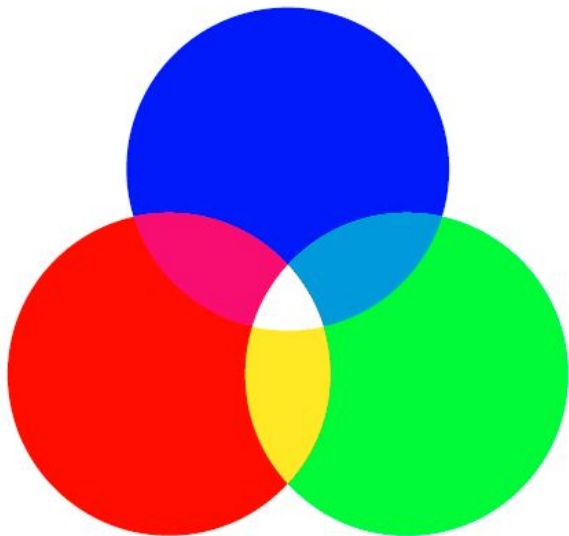
E Color Grading LUT?

Princípio: Utiliza uma *textura LUT* para remapear as cores de uma imagem.

- Ampla variedade de efeitos
- Muito barata computacionalmente
- Artista no controle do resultado final da renderização

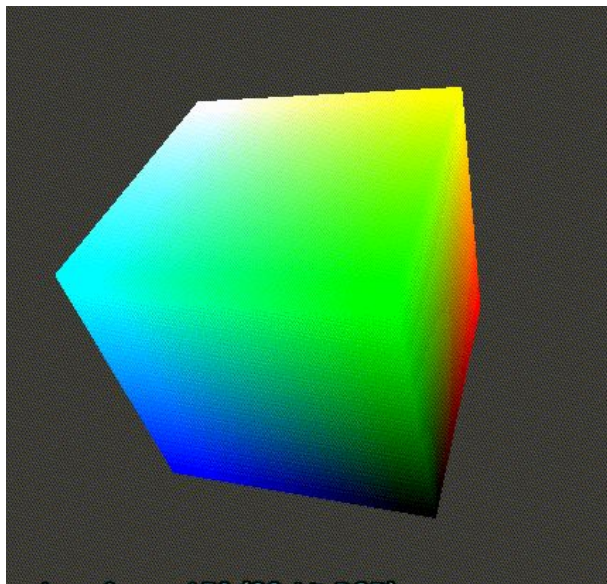
Como funciona o Color Grading LUT?

Padrão RGB



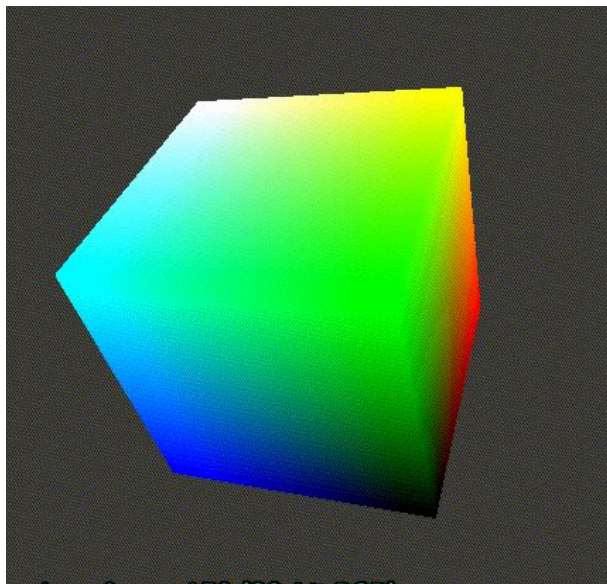
- Sistema de cores aditivas
- Utiliza as cores **vermelho (R)**, **verde (G)** e **azul (B)** para produzir um amplo espectro de cores
- 8 bits: 256 tons para cada cor

Cubo RGB



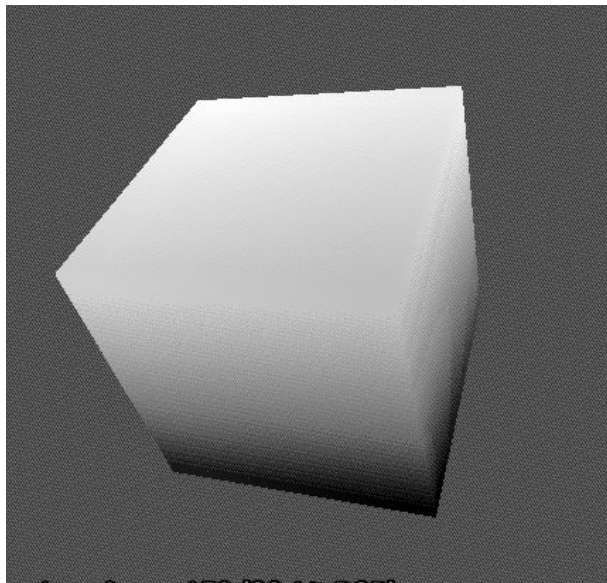
- O espectro de cores RGB pode ser representado através de um *cubo*!
- Cada eixo representa o quanto de vermelho, verde e azul a cor tem
- Todas as cores RGB estão contidas neste cubo

Cubo RGB



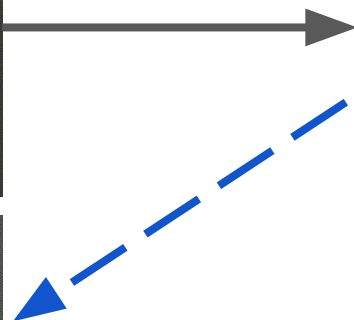
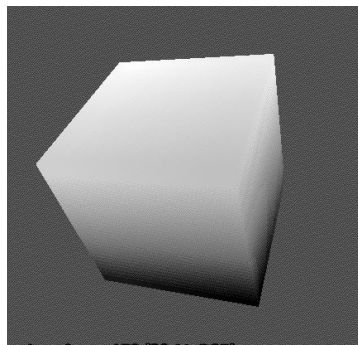
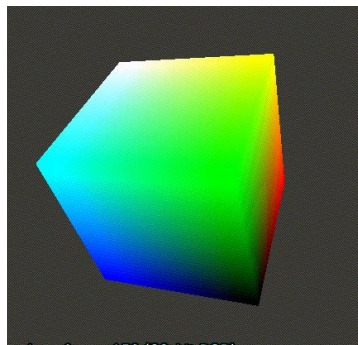
HANDS ON

Gradação de cor

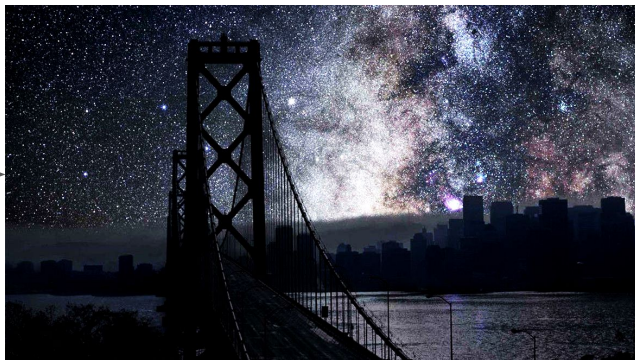
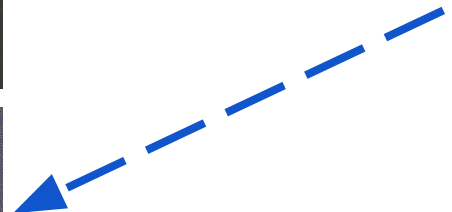
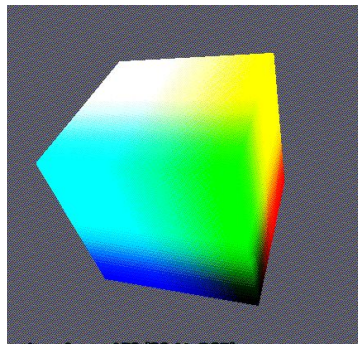
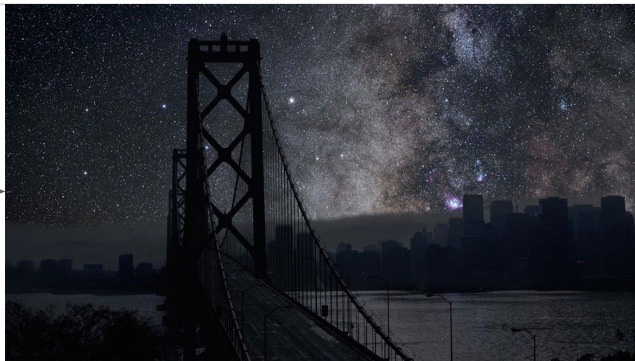
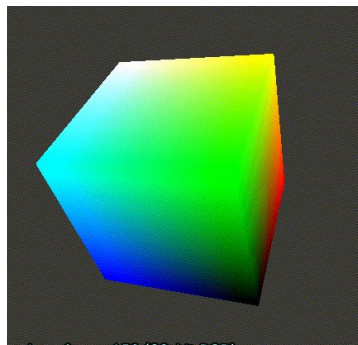


- Podemos mapear as cores de uma imagem para um **outro cubo RGB**
- Nos permite aplicar efeitos interessantes à imagem

Gradação de cor - Exemplo 1



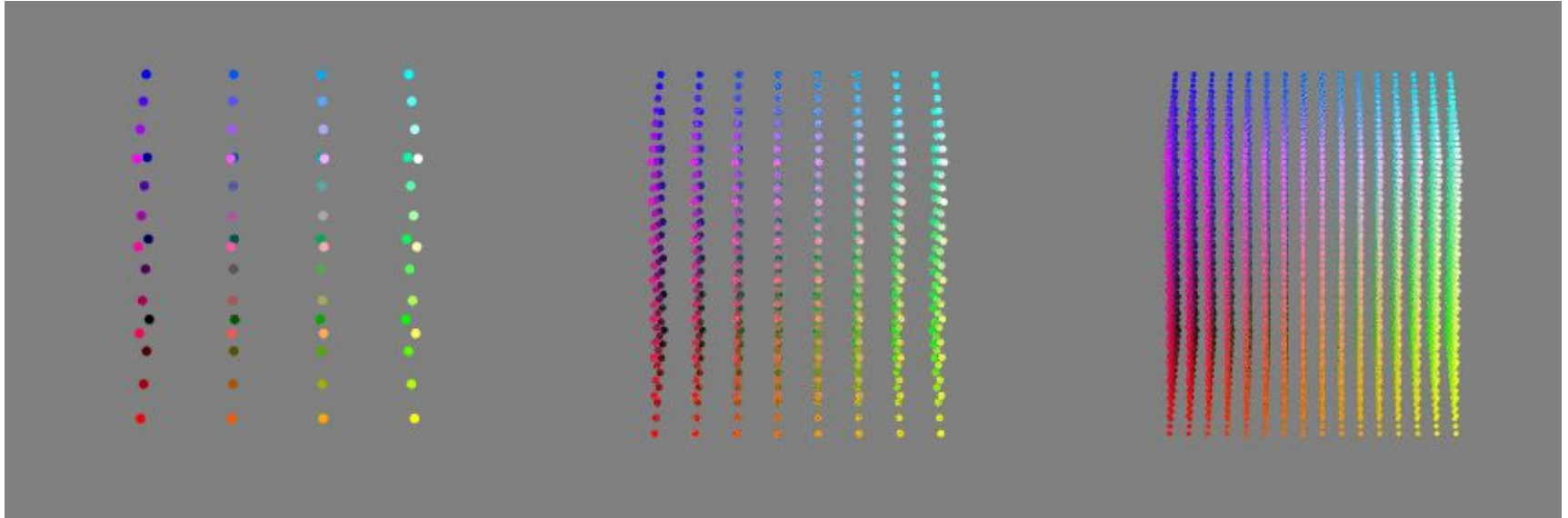
Gradação de cor - Exemplo 2



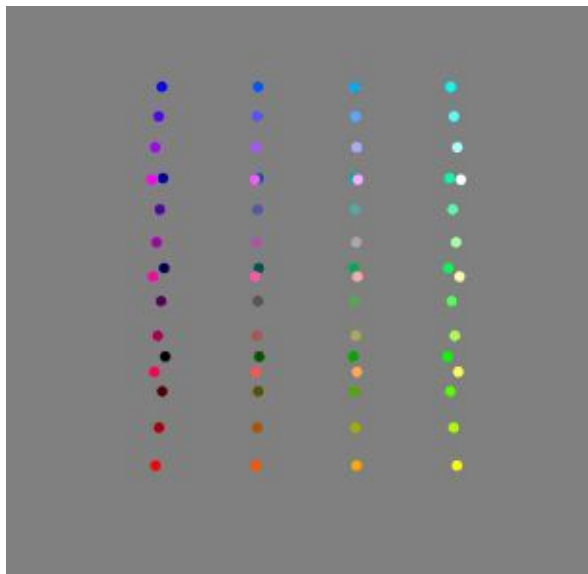
Cubo RGB - Discretização

- **Problema:** Como criar LUTs?
- 8 bits nos canais RGB
 - 256 tons em cada um dos 3 canais = **16.777.216 cores!**
 - Muito pesado para armazenar
- Como guardar todas essas cores?

Cubo RGB - Discretização



Cubo RGB - Discretização



=



Textura Identidade:
4 bits/canal

Look Up Table (LUT)

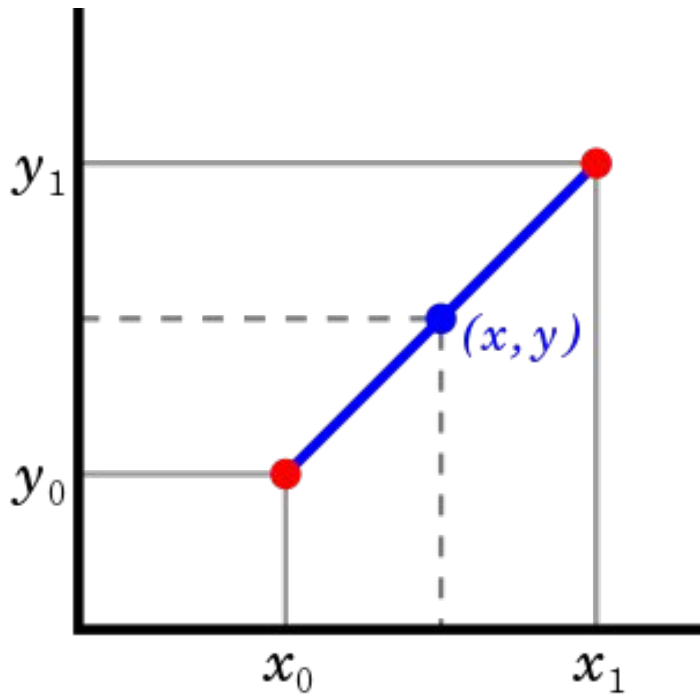


- Textura que mapeia o espaço RGB de maneira discretizada
- Armazena efeitos pré-computados
- Fácil de armazenar e escalável



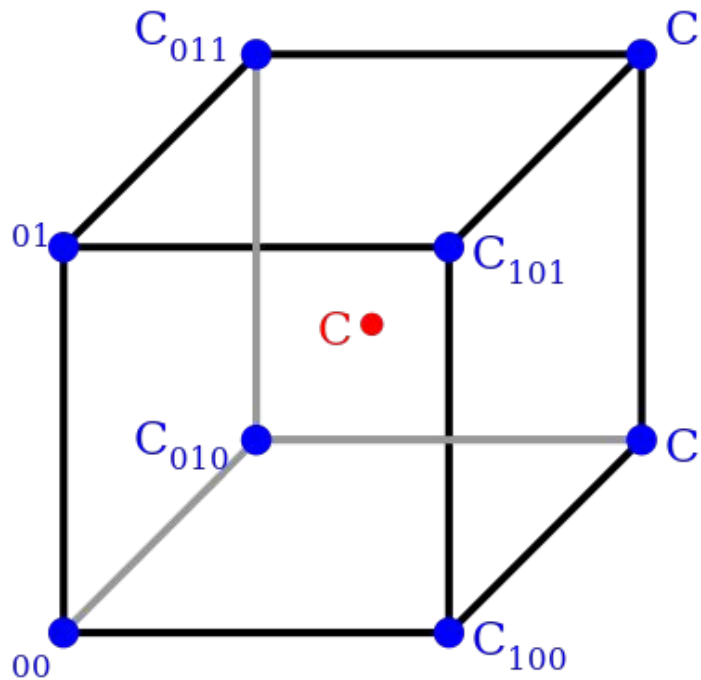
E as outras cores não
presentes na LUT?

Look Up Table - Interpolação



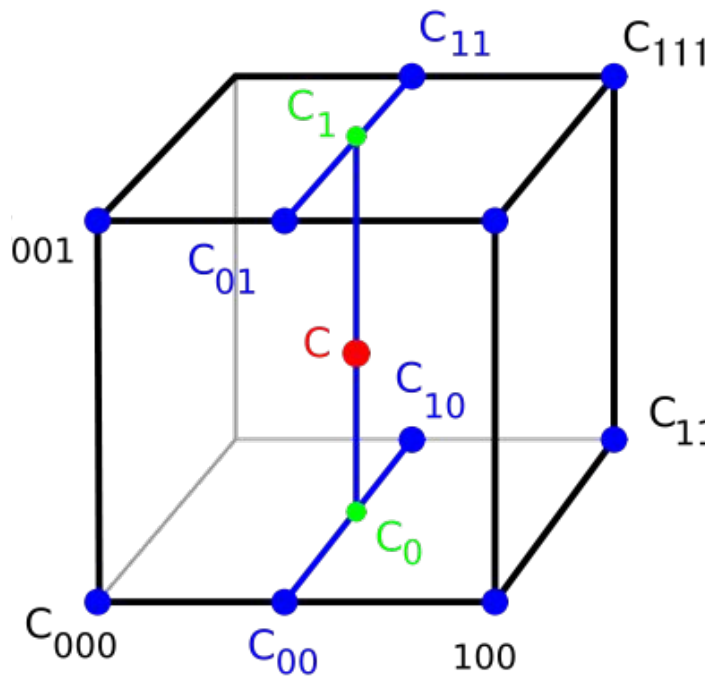
- **Interpolação Linear:** Dadas duas amostras e um offset, cria um valor aproximado linearmente.

Look Up Table - Interpolação



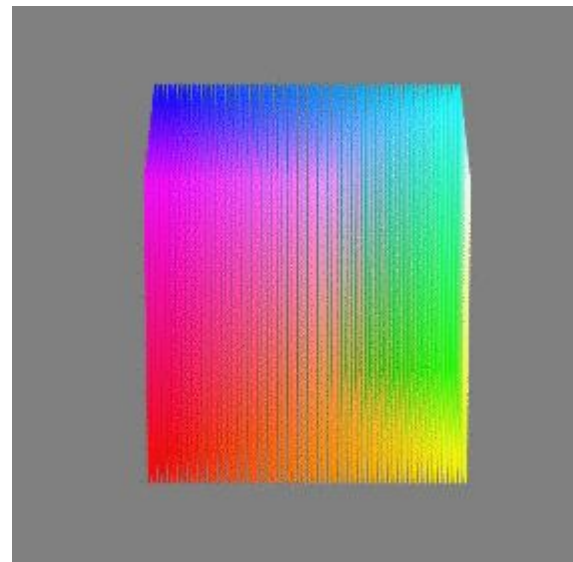
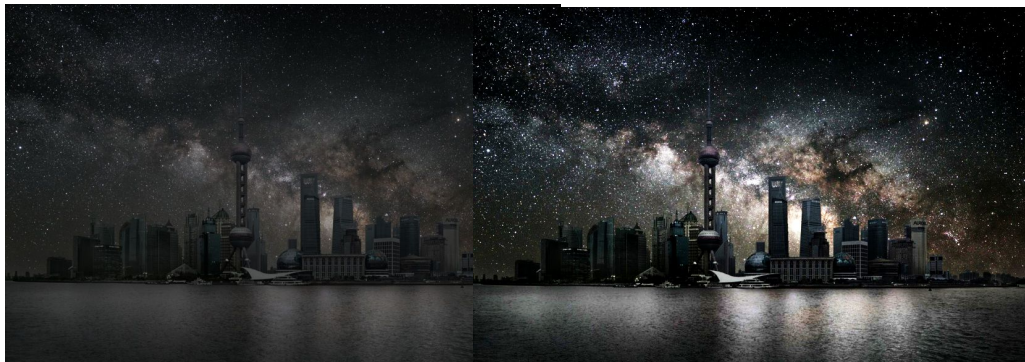
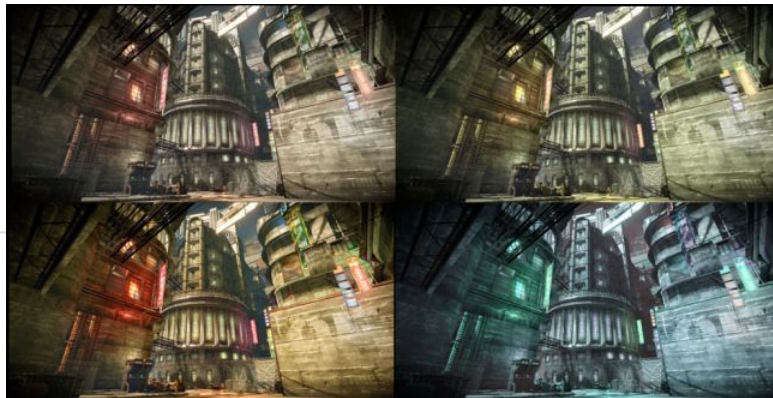
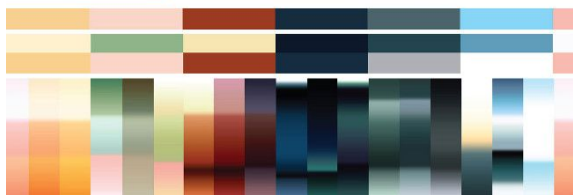
- **Interpolação Trilinear:** utiliza 8 amostras de cores da LUT e um offset nos eixos RGB para criar um valor aproximado linearmente

Look Up Table - Interpolação



- **Interpolação Trilinear:** utiliza 8 amostras de cores da LUT e um offset nos eixos RGB para criar um valor aproximado linearmente

image dump



Como usar Color Grading no meu projeto?

Hands on



GODOT
meetup

Como criar minhas próprias LUTs?

Hands on



GODOT
meetup

Color Grading LUT em tempo real

Hands on



GODOT
meetup

Limitações

- Pixels independentes de posição ou outros pixels
- LUTs em Camadas (*ainda?*)
- Valores precisam ser razoavelmente contínuos
 - LUTs esparsas não interpolam bem
 - *Workaround*: aumentar o tamanho da LUT
 - *Tradeoff*: tamanho da LUT
- Conhecimentos de edição de imagem (criação das LUTs)

Godot Color LUT Shader



[https://github.com/thiagoamendola/
godot-color-lut-shader](https://github.com/thiagoamendola/godot-color-lut-shader)



Perguntas?

amendolathiago@gmail.com

@TicoAmendola





THANK YOU



GODOT
Meetup

São Paulo - SP