

Sobre o Palette Swapper

Há algum tempo atrás, eu estive procurando por um modo de mudar as cores dos sprites na Godot, pois o modulate e o self modulate, inclusos na engine, não eram o que eu precisava para o projeto que estou fazendo... achei vários métodos, todos muito complicados para mim na época.

Até que encontrei um shader que parecia promissor: era pequeno o suficiente para ler entendendo o que estava acontecendo (ou boa parte do que estava acontecendo) e funcional... infelizmente não lembro do autor ou o nome dado ao shader, mas agradeço a ele (ou ela) pela base que me foi fornecida.

Depois de muitos testes e erros, consegui reduzir o shader para o que eu considero, o mínimo necessário para a troca das cores usando paletas de cores em um único arquivo. Não vou mentir, o processo é um pouco trabalhoso, mas compensa.

Este **Palette Swapper** realiza a troca das cores dos sprites de forma eficiente com paletas de cores com até 256 tons de cinza/vermelho e muitos padrões para se utilizar. A explicação também se encontra dentro do arquivo do shader, mas caso você não queira baixar, mas quer, ainda, saber como ele funciona, lá vai:

Requerimentos para o sprite

- Pode ser qualquer imagem, seja em tons de cinza/vermelho ou colorida
- Pode ser spritesheet ou sprite isolado
- O sprite ou spritesheet pode ter qualquer dimensão

Requerimentos para a paleta

- Altura **em potência de 2**. Veja o FAQ para mais informações.
- Pode ter qualquer largura (o shader detecta automaticamente para gerar os índices da paleta)
- Pode ter transparência (total ou parcial)

O shader verifica o canal vermelho da cor do sprite, para determinar qual cor será acessada na paleta.

Tabela para a paleta de cores

IMPORTANTE

O shader SEMPRE lê a paleta de cores de cima para baixo, independente de quantas colunas de cores possua. Veja antes do FAQ deste PDF ou no projeto exemplo como a paleta é estruturada.

Apenas para orientação, fiz esta tabela para mostrar os intervalos de cores e para evitar cálculos desnecessários:

Notas:


- a) Por Cores, entenda "Número de cores usadas no sprite"
- b) Por Tons, entenda "A cada quantos tons de cinza/vermelhos eu separo cada cor para a paleta"

Cores	Tons	Exemplo de uso em jogos conhecidos
2	128	Gato Roboto (Switch - 2 cores)
4	64	Qualquer jogo de Game Boy (Game Boy - 4 cores)
8	32	Super Mario World (SNES - 8 cores por sprite)
16	16	Chrono Trigger (SNES - 15 cores por spritesheet)
32	8	Metal Slug (Arcade - 30 cores por Spritesheet)
64	4	Donkey Kong Country (SNES - 64 cores)
128	2	Ragnarök Online (PC - 98 cores para spritesheets mais complexos)
256	1	Killer Instinct (Arcade - 150+ cores por lutador)

Procurando na internet, em 99% dos casos, os sprites não vem com as cores exatas tanto comparados ao original quanto para o shader (eu mesmo tive esse problema, com os sprites do Mega Man, para montar o exemplo. Achei 6 spritesheets com 6 tons de azul diferentes para o capacete dele). Então como as cores, geralmente, não vem 100% corretas, precisamos ajustá-las.

Qualquer editor de imagens pode realizar esta tarefa, do LibreSprite ao Photoshop, passando pelo Gimp e similares (não estou considerando o Paint porque apesar de ser capaz de alterar o que precisamos ele não trabalhar com transparências).

"Ok, mas o que precisamos ajustar?", você deve estar se perguntando... simples: vamos mudar a cor original, em que o canal vermelho está "errado" para um valor dentro da escala acima. Eu sei que parece complicado, mas é relativamente simples...

 Vamos supor que seu sprite utiliza 2 cores, preto (RGB (0, 0, 0) ou #000000) e cinza (RGB (120, 120, 120) ou #787878).

De acordo com a tabela abaixo:

Cores	Tons	Exemplo de uso em jogos conhecidos
2	128	Gato Roboto (Switch - 2 cores)

Eu preciso que cada cor do meu sprite esteja em um intervalo de 128 tons no canal vermelho para separá-las umas das outras (ora, de 0-127 é uma cor e de 128-255, outra ou em HEX: de 00-7F é uma cor e de 80-FF, outra), lembrando que se o sprite tiver mais cores, os valores mudam (vou deixar uma tabela com um exemplo usando 7 cores, antes do FAQ).

Tabela de organização das cores

Nota: Por "Intervalo", entenda: "intervalo onde a cor se encontra no canal vermelho".

Cores no sprite	Intervalo (HEX)
#000000 ☺	00 - 7F
#787878 ☺	80 - FF

Cores no sprite	Intervalo (RGB)
RGB (0, 0, 0) ☺	0 - 127
RGB (120, 120, 120) ☺	128 - 255



Se você usar o shader sem alterar as cores do sprite, todas as cores ficarão pretas porque, atualmente, o canal vermelho marca 120 (#78....) e esse valor está dentro do intervalo que corresponde ao preto, cujo canal vermelho acessa de 0 a 127 (00 - 7F).

Então basta que, com um editor de imagens, eu substitua a cor RGB (120,120,120) por uma cor próxima como RGB (130,120,120), por exemplo. A diferença visual para nós, humanos, será mínima, praticamente a mesma cor, mas, para o shader, é uma cor completamente diferente. E como o canal vermelho, agora está em 130, o shader consegue reconhecer aquela cor como uma segunda cor.

Cores no sprite	Novas cores do Sprite	Intervalo (HEX)
#000000 ☺	#000000 ☺	00 - 7F
#787878 ☺	#827878 ☺	80 - FF

Cores no sprite	Novas cores do Sprite	Intervalo (RGB)
RGB (0, 0, 0) ☺	RGB (0, 0, 0) ☺	0 - 127
RGB (120, 120, 120) ☺	RGB (130, 120, 120) ☺	128 - 255



Com o sprite com a cor ajustada, vamos para a paleta: no exemplo, agora temos um sprite com as cores preto (RGB (0, 0, 0) ou #000000) e cinza (RGB (130, 120, 120) ou #827878).

Cores no sprite	Novas cores do Sprite	Cores na paleta	Intervalo (HEX)
#000000 ☹️	#000000 ☹️	#000000 ☹️	00 - 7F
#787878 ☹️	#827878 ☹️	#787878 ☹️	80 - FF

Cores no sprite	Novas cores do Sprite	Cores na paleta	Intervalo (RGB)
RGB (0, 0, 0) ☹️	RGB (0, 0, 0) ☹️	RGB (0, 0, 0) ☹️	0 - 127
RGB (120, 120, 120) ☹️	RGB (130, 120, 120) ☹️	RGB (120, 120, 120) ☹️	128 - 255

A paleta será uma imagem com apenas 2px de altura e, para a largura, 1px para cada variação de cor do sprite (lembre-se de colocar as cores originais na paleta, também, para poder voltar para elas a qualquer momento). Então se eu planejo usar um sprite com 5 variações de cor (contando com a original), minha paleta teria 5x2px de dimensões.



Ampliado apenas para visualização

Tabela de exemplo com sprite usando 7 cores em HEX:

Cores do Sprite	Novas cores do Sprite	Cores na paleta	Intervalo (HEX)
#000000 ☹️	#000000 ☹️	#000000 ☹️	00-1F
#FCFCFC ☹️	#FCFCFC ☹️	#FCFCFC ☹️	20-3F
#FC9838 ☹️	#4C9838 ☹️	#FC9838 ☹️	40-5F
#FCD8A8 ☹️	#6CD8A8 ☹️	#FCD8A8 ☹️	60-7F
#80D010 ☹️	#80D010 ☹️	#80D010 ☹️	80-9F
#8000F0 ☹️	#A000F0 ☹️	#8000F0 ☹️	A0-BF
#E40058 ☹️	#CF0058 ☹️	#E40058 ☹️	C0-DF
		#FCFCFC ☹️	E0-FF

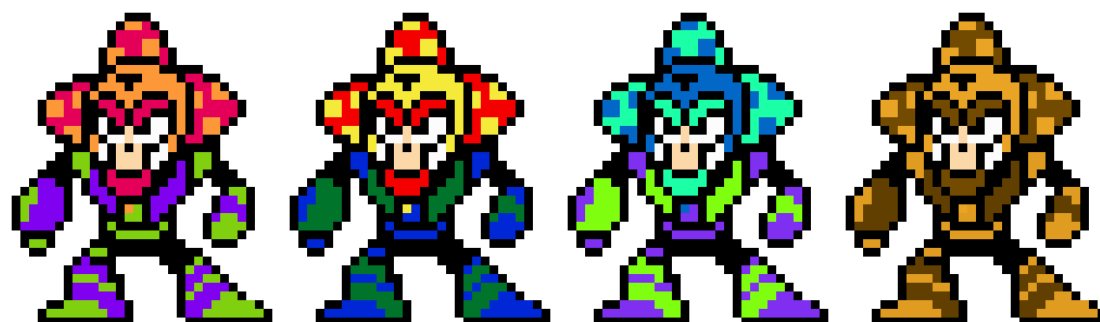
Se você reparar, o branco foi para a última posição na tabela, isso porque se for detectado uma cor que tenha um vermelho que varie de #E0.... até #FF.... ela será substituída pela cor #FCFCFC. “Mas e aquele vazio depois do #000000? Fica assim?” Sim... pois não tem nenhuma cor dentro daquele intervalo. Repare na coluna “Novas cores do Sprite”: não tem nenhuma cor que tenha um vermelho que esteja entre 20 e 3F.

Tabela de exemplo com sprite usando 7 cores em RGB:

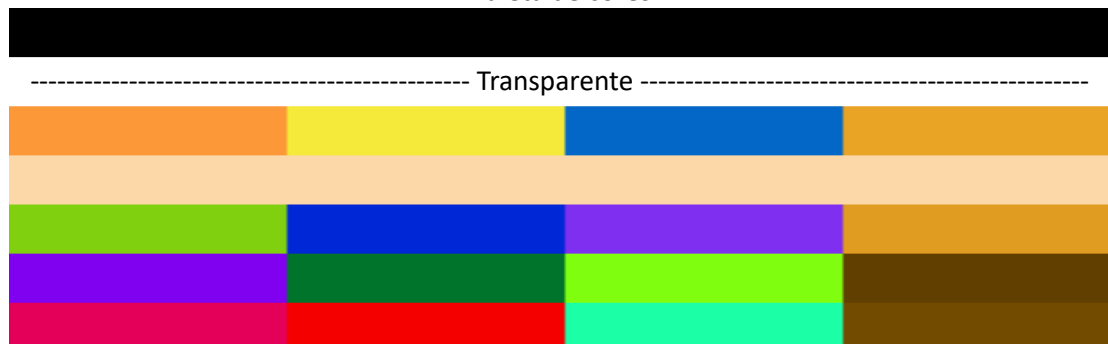
Cores do Sprite	Novas cores do Sprite	Cores na paleta	Intervalo (RGB)
RGB (0, 0, 0) ☹️	RGB (0, 0, 0) ☹️	RGB (0, 0, 0) ☹️	0-31
RGB (252, 252, 252) ☹️	RGB (252, 252, 252) ☹️		32-63
RGB (252, 152, 56) ☹️	RGB (76, 152, 56) ☹️	RGB (252, 152, 56) ☹️	64-95
RGB (252, 216, 168) ☹️	RGB (108, 216, 168) ☹️	RGB (252, 216, 168) ☹️	96-127
RGB (128, 208, 16) ☹️	RGB (128, 208, 16) ☹️	RGB (128, 208, 16) ☹️	128-159
RGB (128, 0, 240) ☹️	RGB (160, 0, 240) ☹️	RGB (128, 0, 240) ☹️	160-191
RGB (228, 0, 88) ☹️	RGB (207, 0, 88) ☹️	RGB (228, 0, 88) ☹️	192-223
		RGB (252, 252, 252) ☹️	224-255

Se você reparar, o branco foi para a última posição na tabela, isso porque se for detectado uma cor que tenha um vermelho que varie de 224 até 255 ela será substituída pela cor RGB (252, 252, 252). “Mas e aquele vazio depois

do RGB (0, 0, 0)? Fica assim?” Sim... pois não tem nenhuma cor dentro daquele intervalo. Repare na coluna “Novas cores do Sprite”: não tem nenhuma cor que tenha um vermelho que esteja entre 32 e 63.



Paleta de cores:



Ampliado apenas para visualização

Como você deve ter notado, junto a este PDF vem também um executável para ver como o shader age e o projeto editável para você pegar o que quiser dele. Espero que tenham gostado da leitura, e lembre-se de ler o FAQ abaixo também.

FAQ

Q: O que são esses canais de que você tanto fala?

A: Forçando a ideia, é um termo usado para indicar alterações em uma parte da cor. Alterar o canal vermelho indica que vou alterar apenas a quantidade de vermelho da cor e deixar o verde e azul como estão.

Q: Você disse anteriormente para usar imagens com altura em potência de 2. Como assim?

A: Disse assim para não escrever várias vezes que as alturas devem ser 2, 4, 8, 16, 32, 64, 128 ou 256 pixels.

Isso se deve, também, ao fato de o cálculo dividir as 256 cores pela altura da paleta, **o que gera sempre um número inteiro** como visto na tabela anteriormente.

Parece limitado, mas dá uma vasta gama de padrões de cores para se trabalhar, levando em conta que a largura é livre.

Q: Minha paleta não tem a altura indicada. O que faço?

A: **Se a altura não for em potência de 2, use o próximo valor (ou o valor anterior) de acordo com a tabela apresentada** (o que não for usado pode ficar transparente ou com uma cor irrelevante). Por exemplo, se meu sprite tem 3 cores, vou usar uma paleta com 4, que é a próxima potência de 2 e a 4ª cor pode ficar transparente, bastando não ter nenhum canal associado a ela. Se tem 17 cores, ou eu removo uma cor, reduzindo para 16 cores ou uso uma paleta de 32 cores, se faço questão de usar todas as cores do sprite.

Q: Meu sprite tem poucas cores. Causa algum problema usar o número exato de cores na paleta?

A: Depende. O shader está configurado para dar o menor trabalho possível de uso. Então se seu sprite tem 3 cores, o cálculo para o índice fica impreciso, gerando dois ou mais índices na mesma cor. Já com 2 ou 4 cores esse problema não ocorre por serem potência de 2.

Q: Meu sprite está com as cores bagunçadas, esse shader não presta!! Não dá pra corrigir isso não?

A: Calma, irmão! Verifique se as cores estão organizadas na paleta e se seu sprite está com as cores obedecendo os padrões que informei no documento acima... Lembre-se que o shader segue uma ordem lógica para fazer o trabalho dele, basta seguir as regras.

Q: Eu quero fazer um holograma simples, posso deixar o sprite meio transparente com esse shader?

A: Sim, basta colocar a cor meio transparente na paleta (recomendo Libresprite, Krita, Gimp ou Photoshop para essa tarefa).

Q: Quero fazer o player sumir parcialmente atrás de uma parede. Esse shader faz isso?

A: Infelizmente, não. Ele serve apenas para mudanças de cor, com ou sem transparências.

Q: Meu jogo tem uma parte que fica tudo escuro, tem como deixar só o contorno branco no sprite?

A: Sim. Veja um exemplo disso lá em cima quando eu explico como organizar as cores do sprite e da paleta: um dos sprites está como contorno branco e coloração preta (poderia ser transparente também).

Q: Eu não sei nada sobre shaders. Eu consigo usar assim mesmo, se precisar?

A: Claro! Tudo o que você vai precisar saber é:

- Como inserir um “material shader” no Godot (para indicar qual a paleta que você vai usar e qual coluna você vai usar para colorir seu sprite)
- Um pouco de GDScript (para alterar a paleta durante o jogo)
- Como criar e editar imagens (editor de sua preferência)
- E, claro, as informações deste pdf (para tirar dúvidas ou relembrar de como o shader funciona)