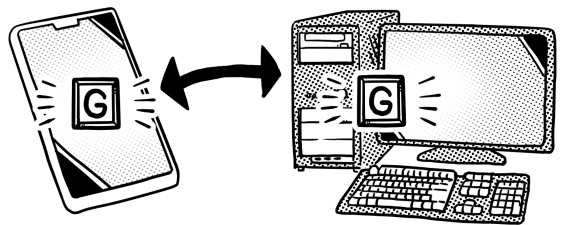


レンダリングパスって何だろう？

漫画: Ikaridon Yu

PCのGPUとモバイルGPUの違いはわかったわ。モバイルGPUアーキテクチャのメリットを生かすにはどうすればいいの？



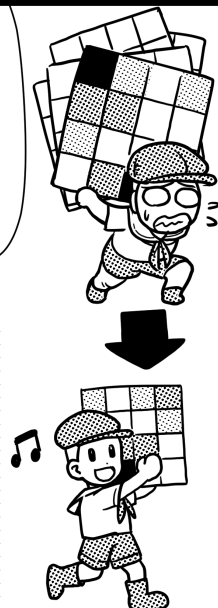
モバイルGPUのデータ転送がバッテリー寿命に左右されることは覚えているかい？



覚えている。だからモバイルGPUには**タイルメモリ**っていうデザインが使われているでしょ…



よろしい。つまりモバイルGPUのメリットを生かすには、転送量を減らさなければならない。

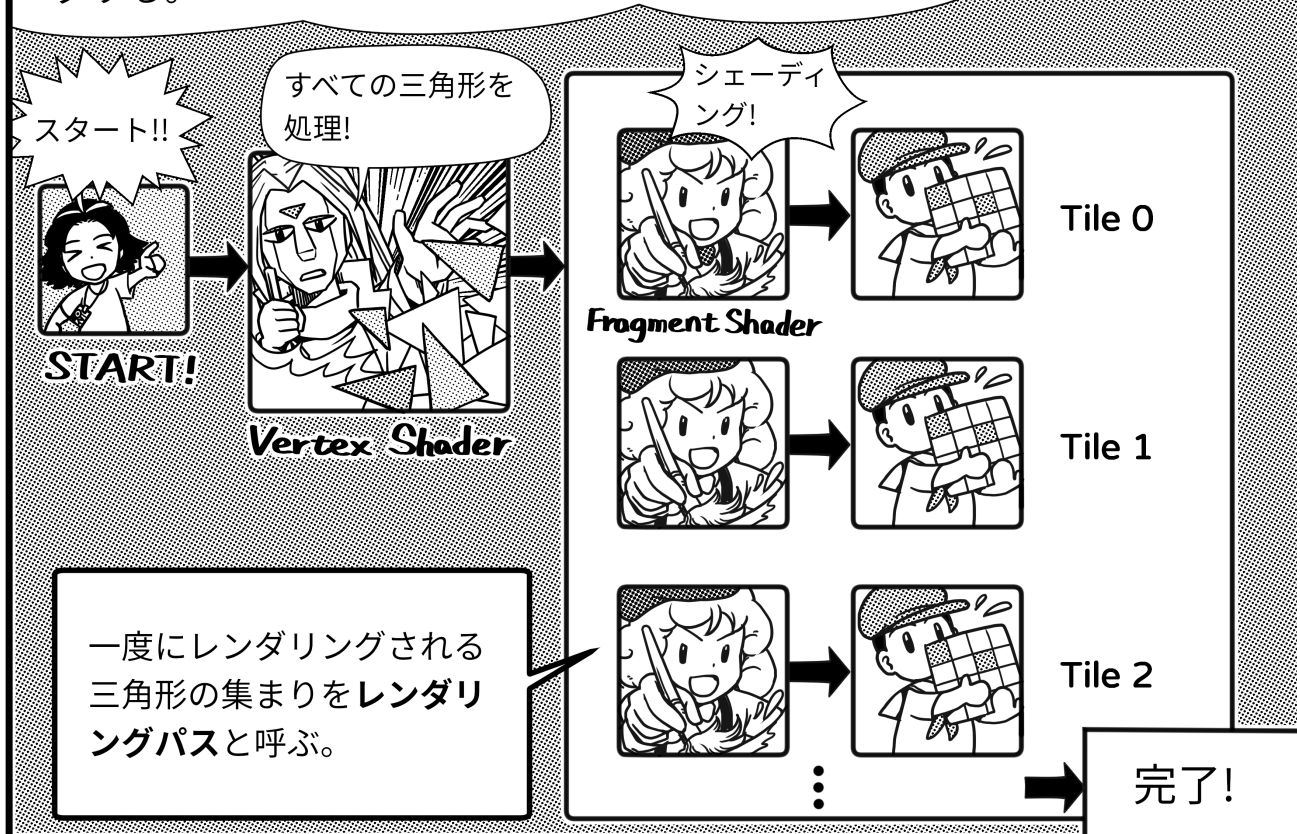


そのためにモバイルGPUには、PCのGPUにはない**レンダリングパス**という特別なデザインが使われているのだよ。



何パス？
チケットみたいなもの？

PCのGPUでは三角形を1つずつレンダリングするが、モバイルGPUでは一度に複数の三角形をレンダリングする。



フレームが終了するまで、タイルはレンダリングパスの中で1つずつ処理される。



注意しないといけないのは、レンダリングパスの最初と最後にやらなければならないタスクが2つあることだ...

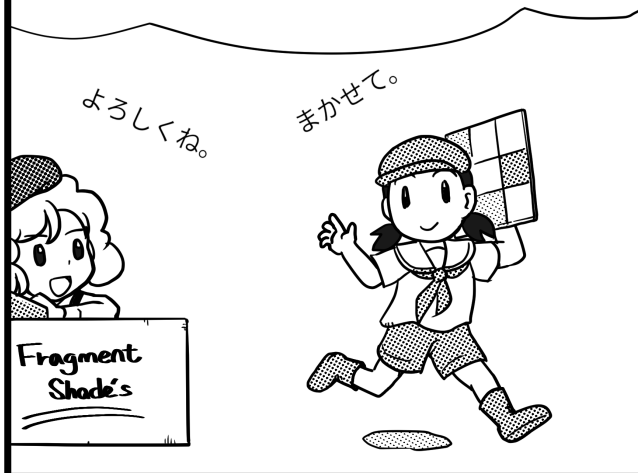


1つめは、レンダリングパスの最初に、最後のフレームのデータをビデオメモリからタイルメモリに移動すること。



データを入れて。色をブレンドするときに必要なだから。

2つめは、レンダリングパスの最後でタイルメモリのデータをビデオメモリに戻すこと。フレームのコンテンツを保持するためだ。



えっ!?

モバイルGPUでそんなことしなかった!なぜゲームが動いたのかしら?



はっはっは。それは...



ぼく、グラフィックスドライバのおかげさ。ぼくがそれをやってあげたんだ。

グラフィックスドライバ

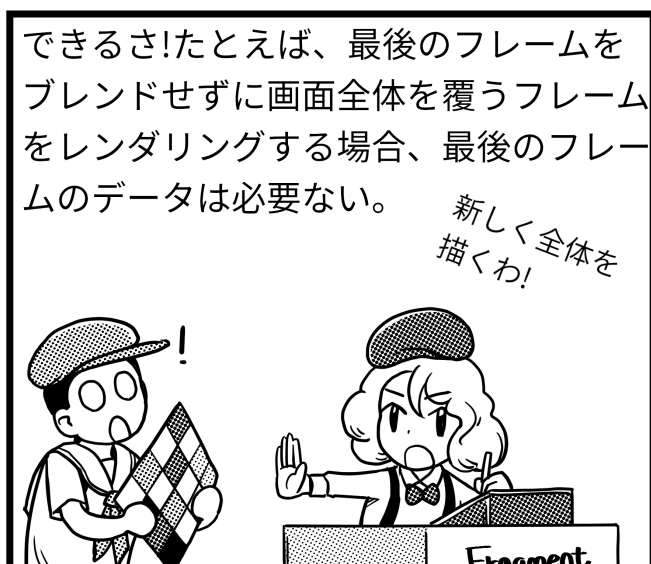


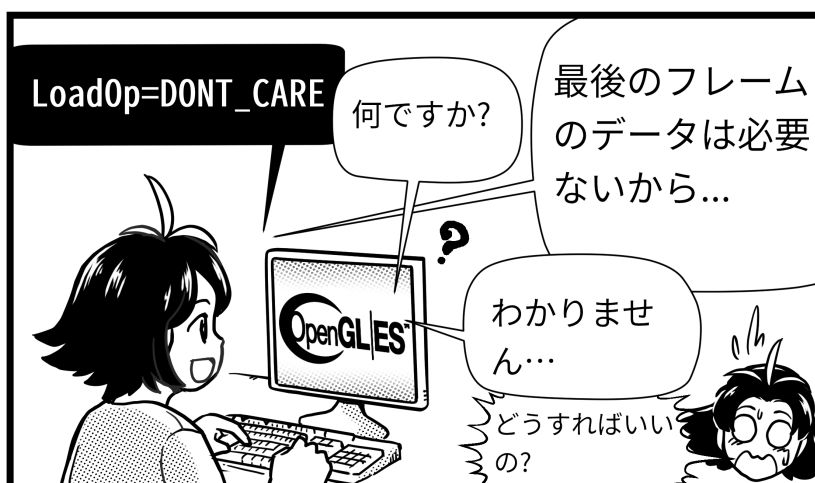
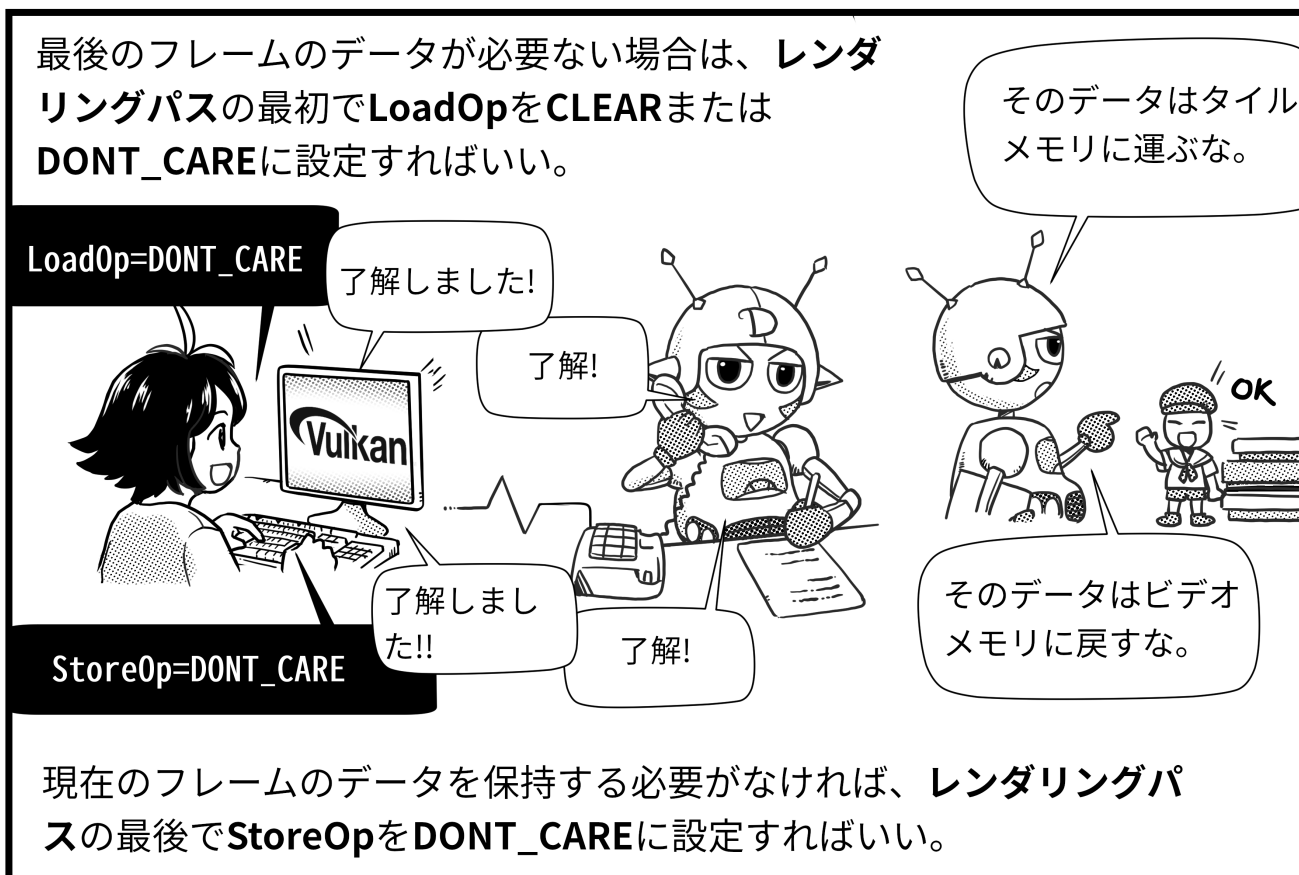
レンダリングパスを作るためにレンダリングする三角形を全部集めたのも僕だよ。



ありがとう、グラフィックスドライバ。あなたがやってくれたのね。







特定のAPIを呼び出してグラフィックスドライバに指示すればいい。

グラフィックスドライバに直接話すわ!

glClear()

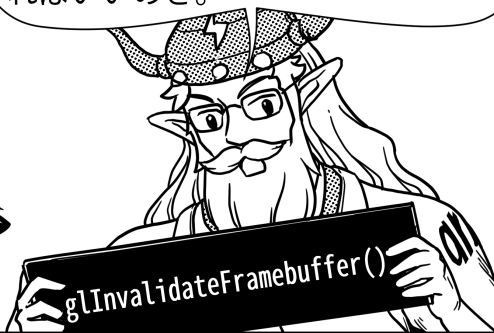
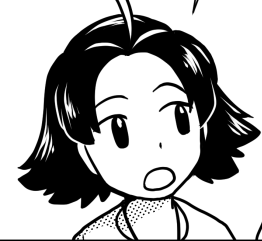
かしこまりました。仰せのとおりいたします。

レンダリングの前に
glClear()か
glInvalidateFramebuffer()
を呼び出して、最後のフレームのデータはいらないとグラフィックスドライバに指示すればいい。

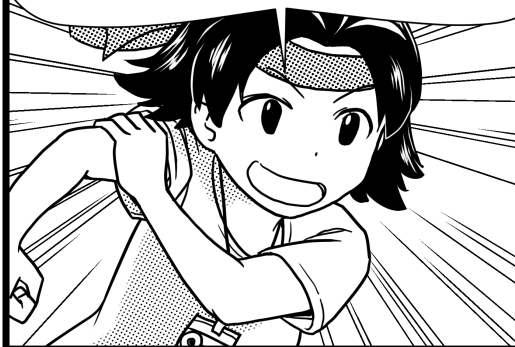


レンダリングが終わったら、どうすればいいの?

glInvalidateFramebuffer()を呼び出して、タイルメモリのデータを捨てるようにドライバに指示すればいいのさ。



わかった!教わった方法でPCゲームをモバイルに移植してみるわ。

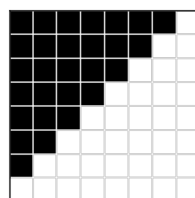


ちょっと待った。最適化のコツがもう1つある。

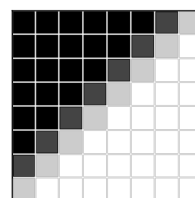
コツ?



今のゲームではアンチエイリアシングを減らすために**MSAA**をよく使うが、**MSAA**を使うと大きな問題が1つある...

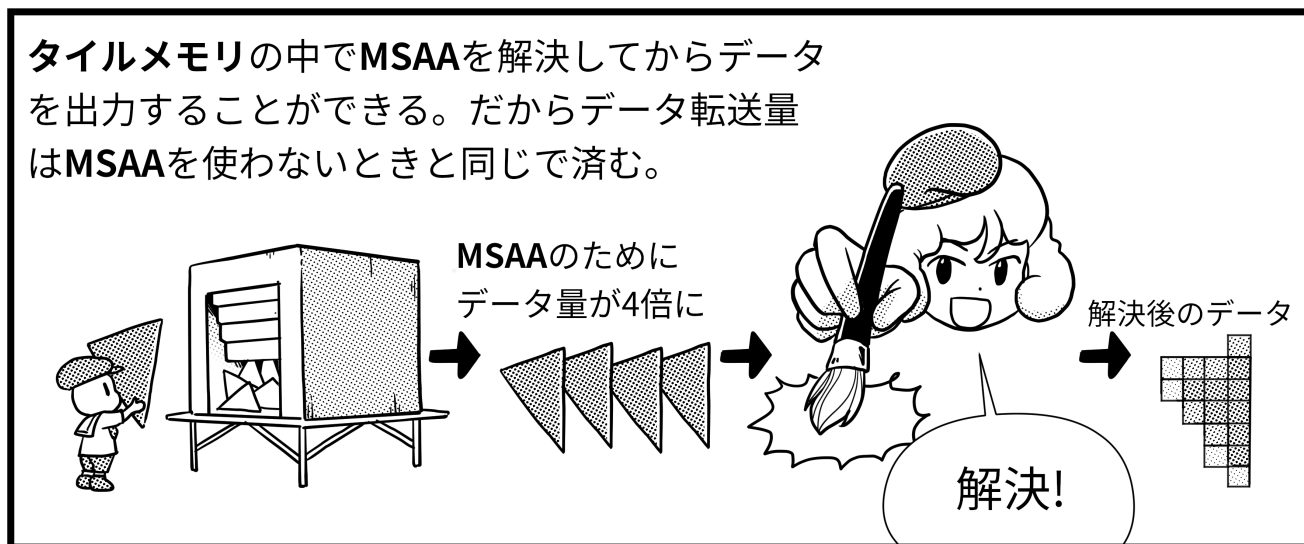


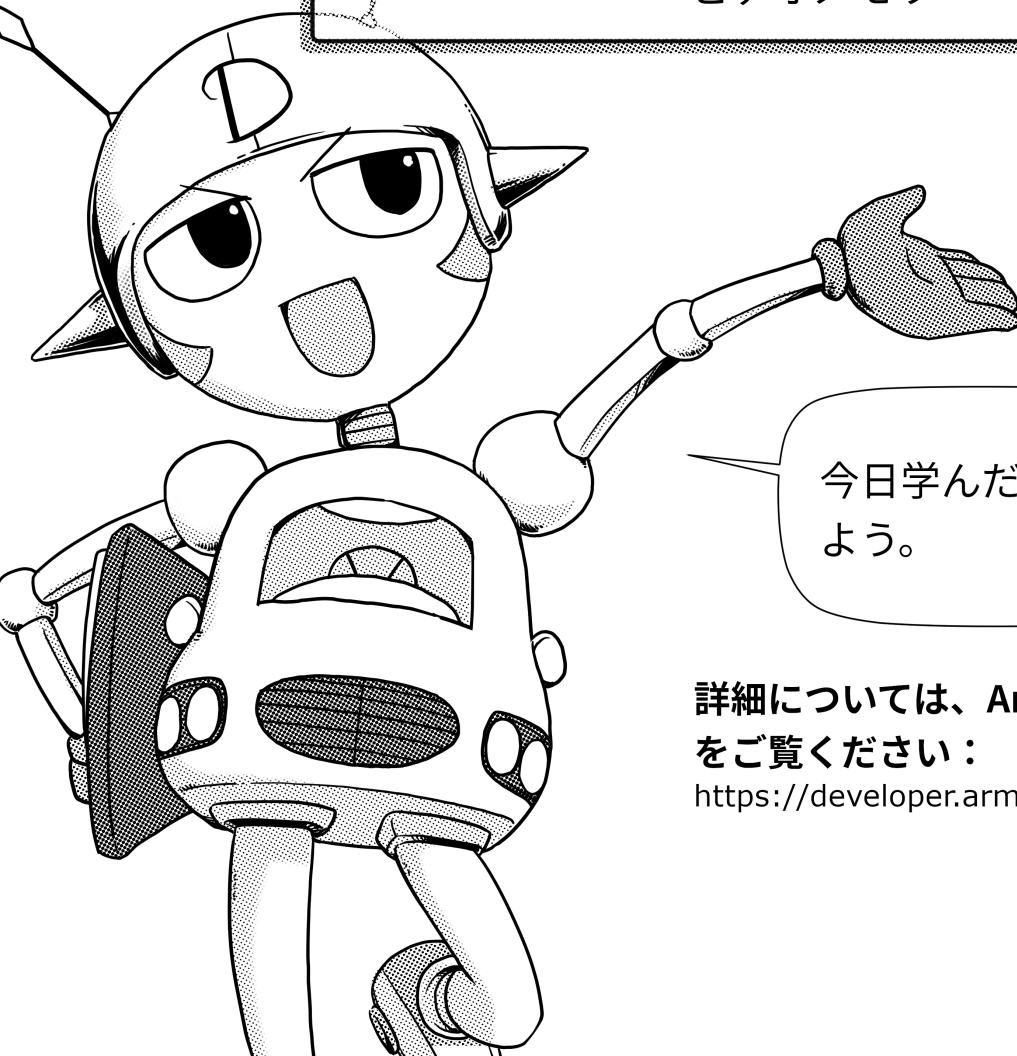
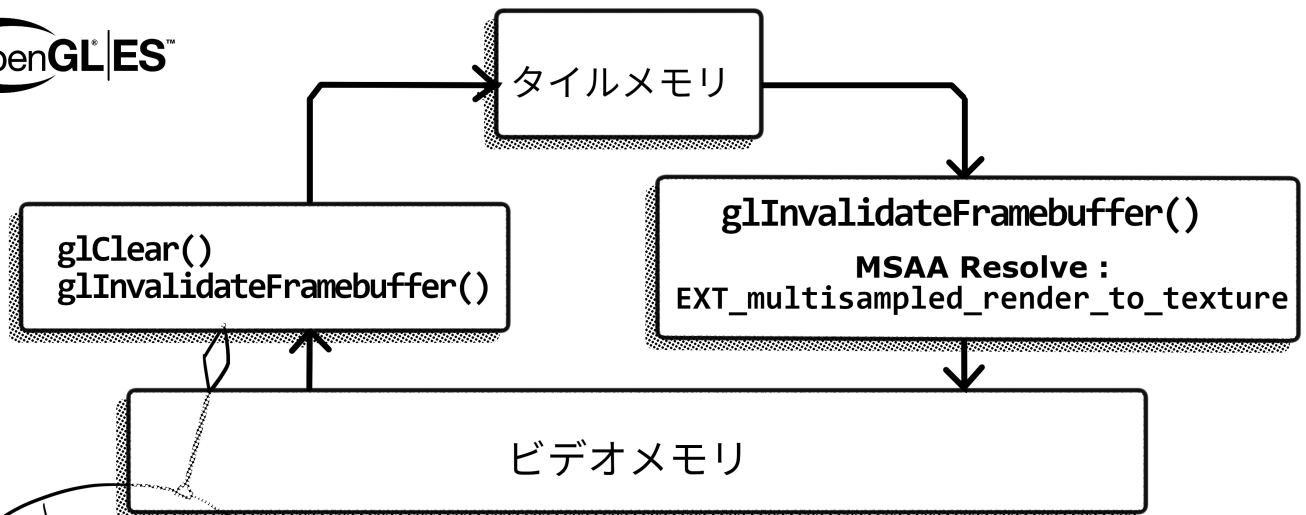
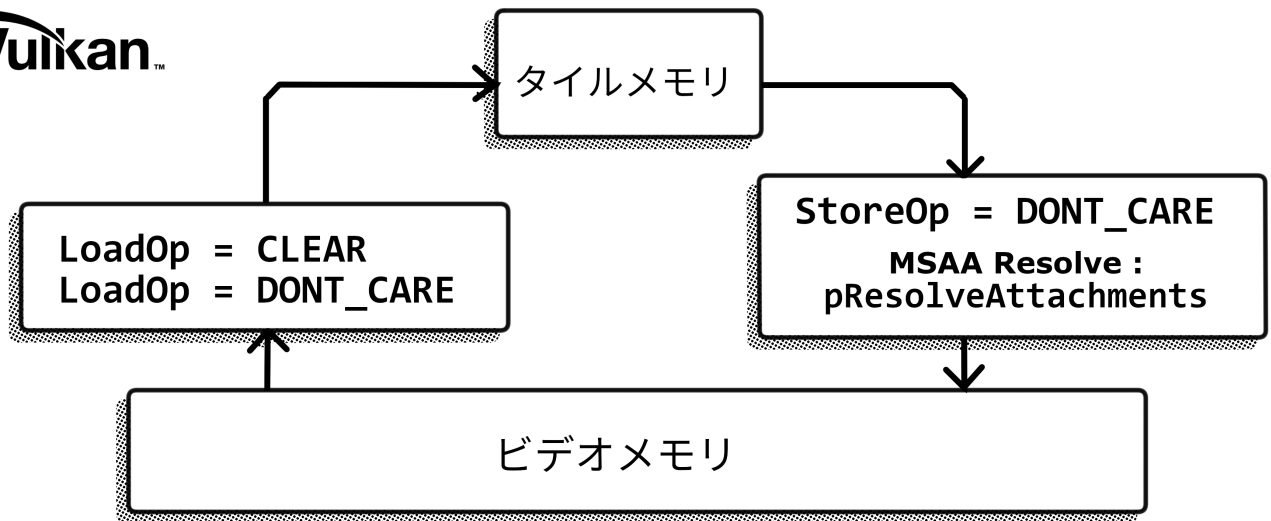
Without Antialiasing



With Antialiasing







今日学んだことを復習しよう。

詳細については、Arm開発者サイトを
ご覧ください：

<https://developer.arm.com/renderpass>