GraphicsSystem立体視について

□GraphicsSystemで立体視を実現する方法には、以下の２つの方法が用意されています。

　１．立体視用バッファを常に確保しておく。

　２．立体視を使いたいときに、立体視用バッファを確保する。

□１の説明

　こちらは、単純で、GraphicsSystemの生成時に渡す、GraphicsSystemSetUp構造体の、

　stereoOnメンバ変数にtrueを渡すことで、いつでも立体視をONにすることが可能になります。

□２の説明

　立体視用に必要な、コマンドリスト２つ（ダブルバッファ用）とフレームバッファ１つをGraphicsSystem生成時ではなく、後で、拡張で確保し、立体視を実現します。

　Kujiraでは、トータルで、0xC1DE0　byteメモリを使用します。

□２の導入手順

　▼立体視をONにするまで

１．②モードで、グラフィックスシステムを生成　（gfl\_use.cppで行っています）

GraphicsSystem( heap::NwAllocator\* allocator, heap::NwAllocator\* deviceAllocator, const GraphicsSystemSetUp\* setup, **bool appUpperRightMode**, u32 systemVramASize = 0x300000, u32 systemVramBSize = 0x300000 );

コンストラクタの、appUpperRightModeにtrueを渡すことで、アプリケーション個別に、立体視バファをON/OFFすることが可能になります。

　２．InitializeAppUpperRightBufferの実行　（アプリケーション単位で行う）

アプリケーション側で、InitializeApplicationMemoryを行った後に、

void InitializeAppUpperRightBuffer( heap::NwAllocator\* deviceAllocator );

　を実行してください。

この関数を実行した瞬間に、コマンドリスト２つとフレームバッファ１つが生成されます。

　３．２フレーム後のレンダリングから立体視になる　（アプリケーション単位で行う）

InitializeAppUpperRightBufferを実行したフレームでは、立体視用のバッファの作成とその仕組みを利用したコマンドの生成を行います。

実際には、次のフレームで、立体視のレンダリング処理が実行され、

２フレーム後に、画面が立体視状態になります。

　▼立体視をOFFにするまで

　１．ReqDeleteAppUpperRightBufferを実行　（アプリケーション単位で行う）

立体視をOFFにするタイミングで、

void ReqDeleteAppUpperRightBuffer( void );

を実行する。

　２．IsDeleteAppUpperRightBufferで、立体視のOFF作業の完了を待つ　（アプリケーション単位で行う）

立体視ように確保した、コマンドリストが確実に使用されていない状態になってから、バッファの削除を行うため、ReqDeleteAppUpperRightBufferをコール後、Delete処理には、２フレーム必要です。

bool IsDeleteAppUpperRightBuffer( void ) const;

で、処理の完了を待ってください。

　３．FinalizeApplicationMemoryを実行　（アプリケーション単位で行う）

以上です。